



Universidad  
Carlos III de Madrid

Trabajo Fin de Grado

# Plataforma de Gestión de Contenidos Web

Francisco Javier Mena Cortés

Grado en Ingeniería de Sistemas Audiovisuales

---

Tutor de empresa: Diego Carrero Figueroa

Tutora: Lourdes Moreno López

Parque tecnológico de Leganés, Julio 2013

Título: Plataforma de Gestión de Contenidos Web

Autor: Francisco Javier Mena Cortés

Directores: Diego Carrero Figueroa y Lourdes Moreno López

EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_ de \_\_\_\_\_ de 20\_\_  
en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid,  
acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# Agradecimientos

Agradezco enormemente la dedicación y el interés de mi tutora Lourdes Moreno que no faltó ni un minuto en todo el proceso de creación y finalización de este proyecto.

Diego Carrero sin duda ha sido la otra mitad de este proyecto, gracias a él he podido aprender, trabajar y vivir el mundo laboral de una manera totalmente amena. Como tutor de proyectos no cabe mayor profesionalidad y como compañero de trabajo no tiene precio.

Finalmente agradezco a mi familia el tiempo de apoyo que me han dado durante todos estos años y los amigos que me han acompañado en este largo viaje como ingeniero.

# Resumen

Ante la evolución y crecimiento en un proyecto web, su plataforma de gestión de contenidos web puede quedarse limitada, y quizás sea necesario desarrollar una extensión de la plataforma web para dar soporte a los nuevos requisitos. Este tipo de desarrollos son peculiares, ya que su diseño y tecnología viene condicionada por el proyecto original.

Éste es el caso del Proyecto Fin de Grado que se presenta, ya que forma parte de la ampliación de la plataforma web “Teatro Accesible” de la empresa Aptent. Por la gran actualización de contenidos de esta plataforma, era esencial dotarla de un gestor de contenidos que proporcionará un entorno de administración y que además, sea sencillo de utilizar por el personal de la empresa Aptent. Esto quiere decir que sin necesidad de tener conocimientos previos de programación o de manejo de base de datos, el personal de la empresa debería ser capaz de publicar, actualizar y/o eliminar contenidos de una forma sencilla. Dicha plataforma se está utilizando en la actualidad con éxito y satisfacción por parte de la empresa Aptent.

El desarrollo de este gestor de contenidos siguiendo un enfoque metodológico según requisitos definidos por la empresa Aptent es la principal aportación de este PFG. Previo a este desarrollo ha sido necesario llevar un estudio de la aplicación original y su dominio, su arquitectura y diseño, así como de las tecnologías con las cuáles poder crecer y llevar a cabo la ampliación solicitada.

**Palabras clave:** gestor de contenidos web, teatro accesible, CMS, Aptent, modelo-vista-controlador, PHP, MySQL, , Yii Framework.



# Abstract

Given the evolution and growth in a web project, the content management platform project website can be limited, and it may be necessary to develop a web platform continued to support the new requirements. Such developments are peculiar, because its design and technology is conditioned by the original project.

This is the case of the Final Grade Project (PFG) presented, on account of it is a part of the expansion of the Aptent company web platform "Teatro Accesible". For the big content update, it was essential to provide to this environment with a content manager system that will provide a management environment and also be easy to use by company staff Aptent. This means that without having prior knowledge of programming or database management, the staff of the company should be able to publish, update and/or remove content in a simple way. This platform is being used successfully today and satisfaction of the company Aptent.

The development of the content management system following a methodological approach as defined by the Aptent's business requirements and is the main contribution of this PFG. Prior to this development has been necessary a study of the original application and its domain, its architecture and design, as well as the technologies with which to grow and carry out the requested extension.

**Keywords:** web content management, accessible theater, CMS, Aptent, model-view-controller, PHP, MySQL, Yii Framework.





# Índice general

- 1. Introducción y objetivos ..... 19
  - 1.1 Introducción ..... 19
  - 1.2 Objetivos..... 20
  - 1.3 Fases del desarrollo..... 20
  - 1.4 Medios empleados..... 20
  - 1.5 Estructura de la memoria..... 20
- 2. Estado del Arte..... 22
  - 2.1 Introducción ..... 22
  - 2.2 Antecedentes del proyecto ..... 23
    - 2.2.1 Dominio..... 23
    - 2.2.2 Aspectos socioeconómicos..... 24
  - 2.3 Tecnologías de desarrollo de aplicaciones web..... 25
  - 2.4 Categorías de Aplicaciones web ..... 27
  - 2.5 Gestor de contenidos..... 31
  - 2.6 Trabajos relativos..... 32
  - 2.7 Discusión de los datos y propuesta de solución..... 33
- 3. Estudio tecnológico..... 34
  - 3.1 Introducción ..... 34
  - 3.2 Herramientas de desarrollo ..... 34
    - 3.2.1 HTML..... 34
    - 3.2.2 JavaScript..... 35
    - 3.2.3 PHP..... 36
    - 3.2.4 Sistema Gestor de Base de Datos: MySQL..... 37
    - 3.2.5 Yii Framework..... 37

3.3	Tecnología entorno de desarrollo.....	38
3.3.1	APACHE Server .....	38
3.3.2	Aptana Studio 3.0.....	39
3.3.3	Xampp.....	39
3.3.4	TeamLab .....	39
3.3.5	Google Chrome.....	40
4.	Análisis.....	41
4.1	Introducción .....	41
4.2	Requisitos impuestos y arquitectura del sistema .....	41
4.3	Arquitectura de la Información .....	45
4.3.1	Precedentes.....	45
4.3.2	Decisión de la arquitectura de información.....	47
4.4	Catálogo de requisitos .....	48
4.5	Casos de uso .....	49
5.	Diseño e implementación.....	52
5.1	Introducción .....	52
5.2	Diseño de clases .....	53
5.2.1	Modelos.....	53
5.2.2	Vistas.....	65
5.2.3	Controlador .....	56
5.2.4	Layouts.....	57
5.2.5	Widgets .....	57
5.3	Implementación de la aplicación web.....	58
6.	Pruebas .....	63
6.1	Introducción .....	63
6.2	Pruebas unitarias.....	63
6.3	Pruebas de integración.....	64
6.4	Catálogo de pruebas .....	64
7.	Manual de usuario.....	66
7.1	Introducción .....	66
7.2	Instalación .....	66
7.3	Acceso.....	66
7.4	Utilización .....	67
7.5	Cierre de sesión.....	70
8.	Gestión del proyecto.....	71
8.1	Introducción .....	71
8.2	Metodología .....	72

8.3 Estimación de recursos ..... 73

8.4 Estimación de recursos económicos..... 75

    8.4.1 Recursos Materiales..... 75

    8.4.2 Recursos Humanos..... 75

    8.4.3 Costes Totales..... 76

9. Conclusiones y líneas futuras ..... 77

9.1 Introducción ..... 77

9.2 Conclusiones..... 77

9.3 Líneas futuras..... 78

10. Referencias ..... 77

# Índice de figuras

Figura 2.1: Iconos de accesibilidad (Aptent)

Figura 2.2: Modelo cliente servidor.

Figura 2.3: Arquitectura de tres capas

Figura 2.4: Clasificación de aplicaciones.

Figura 2.5: Evolución de la Web

Figura 2.6: Modelo clásico de aplicación web

Figura 2.7: Modelo MVC de una aplicación web

Figura 3.1: Estadísticas lenguajes de programación más visitados del lado del cliente

Figura 3.2: Estadísticas de uso de lenguajes de programación de servidores

Figura 3.3: Gráfica comparativa de frameworks en PHP

Figura 4.1: Modelo MVC por capas

Figura 4.2: Estructura estática de aplicación Yii

Figura 4.3: Ciclo de vida de una petición

Figura 4.4: Arquitectura del proyecto.

Figura 4.5: Opciones del menú de Teatro Accesible.

Figura 4.6: Captura de pantalla de la base de datos.

Figura 4.7: Mapa global de Casos de Uso

Figura 5.1: Diagrama de paquetes del proyecto para la arquitectura MVC seleccionada

Figura 5.2: Mapa del la arquitectura de clases modelo.

Figura 5.3: Vista principal del CMS.

Figura 5.4: Menú de elementos.

Figura 5.5: Tabla de atributos del elemento seleccionado.

Figura 5.6: Formulario para la creación de un nuevo elemento.

Figura 5.7: Formulario para editar un elemento.

Figura 5.8: Evento de confirmación para eliminar.

Figura 5.9: Pantalla de detalles del elemento seleccionado

Figura 5.10. Página de autenticación.

Figura 7.1. Formulario de entrada a la aplicación.

Figura 7.2. Menú de selección de elemento,

Figura 7.3. Listado de elementos de la base de datos.

Figura 7.4. Menú de opciones.

Figura 7.5. Formulario de creación de un elemento.

Figura 7.6. Sistema de selección de elementos.

Figura 7.7. Formulario de edición de un elemento.

Figura 7.8. Ventanas emergentes de alertas para la eliminación de elementos.

Figura 7.9. Enlace a los detalles de un elemento.

Figura 7.10: Texto de cierre de sesión.

Figura 8.1. Proceso de Desarrollo Software

Figura 8.2. Ciclo de Vida en Espiral

Figura 8.3. Diagrama de Gantt

Figura 8.4. Expresión para el Cálculo de las Amortizaciones

# Índice de tablas

- Tabla 1: Tabla genérica de requisitos de datos
- Tabla 2: Tabla genérica de requisitos
- Tabla 3: Tabla genérica de evaluación de pruebas
- Tabla 4: Tabla genérica de casos de uso
- Tabla 5: Recursos Temporales por Fases del Proyecto
- Tabla 6: Recursos Materiales para el Desarrollo del Proyecto.
- Tabla 7: Recursos Humanos.

**Tablas ANEXO I**

- Tabla 1: Requisito de datos REQ.D.001: Información sobre Lenguajes
- Tabla 2: Requisito de datos REQ.D.002: Información sobre Teatros
- Tabla 3: Requisito de datos REQ.D.003: Información sobre Ciudades
- Tabla 4: Requisito de datos REQ.D.004: Información sobre Temporadas
- Tabla 5: Requisito de datos REQ.D.005: Información sobre Producciones
- Tabla 6: Requisito de datos REQ.D.006: Información sobre Performance
- Tabla 7: Requisito de datos REQ.D.007: Información sobre Portadas
- Tabla 8: Requisito de datos REQ.D.008: Información sobre Links
- Tabla 9: Requisito de datos REQ.D.009: Información sobre Noticias
- Tabla 10: Requisito de datos REQ.D.010: Información sobre Galerías
- Tabla 11: Requisito de datos REQ.D.011: Información sobre Contenido
- Tabla 12: Requisito de datos REQ.D.012: Información sobre Sesión
- Tabla 13: Requisito de aplicación REQ.A.001: Plataforma Hardware
- Tabla 14: Requisito de aplicación REQ.A.002: Consistencia en la Base de datos
- Tabla 15: Requisito de aplicación REQ.A.003: Servicio de hosting STRATO

Tabla 16: Requisito de aplicación REQ.A.004: Lenguaje PHP

Tabla 17: Requisito de aplicación REQ.A.005: Base de datos en MySQL

Tabla 18: Requisito de aplicación REQ.A.006: Plataforma de trabajo Yii Framwork

Tabla 19: Requisito de aplicación REQ.A.007: Entorno de desarrollo Aptana Studio

Tabla 20: Requisito de interfaz REQ.I.001: Perfil de Administrador

Tabla 21: Requisito de interfaz REQ.I.002: Inicio de sesión

Tabla 22: Requisito de interfaz REQ.I.003: Cierre de sesión

Tabla 23: Requisito de interfaz REQ.I.004: Usabilidad de controles y vistas

Tabla 24: Requisito de funcionalidad REQ.F.001: Gestión de contenido de programación

Tabla 25: Requisito de funcionalidad REQ.F.002: Gestión de contenido adicional

Tabla 26: Requisito de funcionalidad REQ.F.003: Gestión de Idiomas

Tabla 27: Requisito de funcionalidad REQ.F.001.1: Añadir nueva Ciudad

Tabla 28: Requisito de funcionalidad REQ.F.001.2: Editar Ciudad

Tabla 29: Requisito de funcionalidad REQ.F.001.3: Añadir nuevo Teatro

Tabla 30: Requisito de funcionalidad REQ.F.001.4: Editar Teatro

Tabla 31: Requisito de funcionalidad REQ.F.001.5: Añadir nueva Temporada

Tabla 32: Requisito de funcionalidad REQ.F.001.6: Editar Temporada

Tabla 33: Requisito de funcionalidad REQ.F.001.7: Añadir nueva Producción

Tabla 34: Requisito de funcionalidad REQ.F.001.8: Editar Producción

Tabla 35: Requisito de funcionalidad REQ.F.001.9: Añadir nueva Performance

Tabla 36: Requisito de funcionalidad REQ.F.001.10: Editar Performance

Tabla 37: Requisito de funcionalidad REQ.F.001.11: Añadir nueva Sesión

Tabla 38: Requisito de funcionalidad REQ.F.001.12: Editar Sesión

Tabla 39: Requisito de funcionalidad REQ.F.002.1: Añadir nueva Galería

Tabla 40: Requisito de funcionalidad REQ.F.002.2: Editar Galería

Tabla 41: Requisito de funcionalidad REQ.F.002.3: Eliminar Galería

Tabla 42: Requisito de funcionalidad REQ.F.002.4: Añadir nueva Noticia

Tabla 43: Requisito de funcionalidad REQ.F.002.5: Editar Noticias

Tabla 44: Requisito de funcionalidad REQ.F.002.6: Eliminar Noticias

Tabla 45: Requisito de funcionalidad REQ.F.002.7: Añadir nuevo Link

Tabla 46: Requisito de funcionalidad REQ.F.002.8: Editar Link

Tabla 47: Requisito de funcionalidad REQ.F.002.9: Eliminar Link

Tabla 48: Requisito de funcionalidad REQ.F.002.10: Añadir nueva Portada

Tabla 49: Requisito de funcionalidad REQ.F.002.11: Añadir nueva Portada

Tabla 50: Requisito de funcionalidad REQ.F.002.12: Eliminar Portada

Tabla 51: Requisito de funcionalidad REQ.F.002.13: Añadir nuevo Contenido

Tabla 52: Requisito de funcionalidad REQ.F.002.14: Editar Contenido  
Tabla 53: Requisito de funcionalidad REQ.F.002.15: Eliminar Contenido  
Tabla 54: Requisito de funcionalidad REQ.F.003.1: Añadir nuevo Lenguaje  
Tabla 55: Requisito de funcionalidad REQ.F.003.2: Editar Lenguaje  
Tabla 56: Requisito de rendimiento REQ.R.001: Tiempo de Respuesta  
Tabla 57: Requisito de rendimiento REQ.R.002: Carga computacional  
Tabla 58: Requisito de seguridad REQ.S.001: Almacenamiento de contraseñas  
Tabla 59: Requisito de seguridad REQ.S.002: Seguridad del servidor web

## **Tablas ANEXO II**

Tabla 1: C.U.001: Autenticación de usuario  
Tabla 2: C.U.002: Gestión de contenido de programación  
Tabla 3: C.U.013: Gestión de contenido adicional  
Tabla 4: C.U.026: Gestión de idiomas  
Tabla 5: C.U.029: Cierre de sesión  
Tabla 6: C.U.003: Añadir Ciudad  
Tabla 7: C.U.004: Editar Ciudad  
Tabla 8: C.U.005: Añadir Teatro  
Tabla 9: C.U.006: Editar Teatro  
Tabla 10: C.U.007: Añadir Temporada  
Tabla 11: C.U.008: Editar Temporada  
Tabla 12: C.U.009: Añadir Producción  
Tabla 13: C.U.010: Editar Producción  
Tabla 14: C.U.011: Añadir Performance  
Tabla 15: C.U.012: Editar Performance  
Tabla 16: C.U.012.1: Añadir Sesión  
Tabla 17: C.U.012.2: Editar Sesión  
Tabla 18: C.U.014: Añadir Link  
Tabla 19: C.U.015: Editar Link  
Tabla 20: C.U.016: Eliminar Link  
Tabla 21: C.U.020: Añadir Noticia  
Tabla 22: C.U.021: Editar Noticia  
Tabla 23: C.U.022: Eliminar Noticia  
Tabla 24: C.U.023: Añadir Portada  
Tabla 25: C.U.024: Editar Portada  
Tabla 26: C.U.025: Eliminar Portada  
Tabla 27: C.U.017: Añadir Galería



Tabla 28: C.U.018: Editar Galería  
Tabla 29: C.U.019: Eliminar Galería  
Tabla 30: C.U.018.1: Añadir Contenido  
Tabla 31: C.U.018.2: Editar Contenido  
Tabla 32: C.U.018.3: Eliminar Contenido  
Tabla 33: C.U.027: Añadir Lenguaje  
Tabla 34: C.U.028: Editar Lenguaje

### **Tablas ANEXO III**

Tabla 1: P.001-Verificación del acceso a la aplicación  
Tabla 2: P.002-Comprobación de la correcta creación del elemento ciudad  
Tabla 3: P.003- Comprobación de la correcta creación del elemento teatro  
Tabla 4: P.004- Comprobación de la correcta creación del elemento temporada  
Tabla 5: P.005- Comprobación de la correcta creación del elemento producción  
Tabla 6: P.006- Comprobación de la correcta creación del elemento performance  
Tabla 7: P.007- Comprobación de la correcta creación del elemento enlace  
Tabla 8: P.008- Comprobación de la correcta creación del elemento noticias  
Tabla 9: P.009- Comprobación de la correcta creación del elemento portada  
Tabla 10: P.0010- Comprobación de la correcta creación del elemento galería  
Tabla 11: P.0011- Comprobación de la correcta creación del elemento idioma  
Tabla 12: P.0012- Verificación del cierre de sesión de la aplicación  
Tabla 13: P.0013 - Comprobación de la correcta edición del elemento ciudad  
Tabla 14: P.0014 - Comprobación de la correcta edición del elemento teatro  
Tabla 15: P.0015- Comprobación de la correcta edición del elemento temporada  
Tabla 16: P.0016- Comprobación de la correcta edición del elemento producción  
Tabla 17: P.0017- Comprobación de la correcta edición del elemento performance  
Tabla 18: P.0018- Comprobación de la correcta edición del elemento enlace  
Tabla 19: P.0019- Comprobación de la correcta edición del elemento noticias  
Tabla 30: P.0020- Comprobación de la correcta edición del elemento portada  
Tabla 31: P.0021- Comprobación de la correcta edición del elemento galería  
Tabla 32: P.0022- Comprobación de la correcta edición del elemento idioma  
Tabla 33: P.0023- Comprobación de la correcta eliminación del elemento enlace  
Tabla 34: P.0024- Comprobación de la correcta eliminación del elemento noticia  
Tabla 35: P.0025- Comprobación de la correcta eliminación del elemento portada  
Tabla 36: P.0026- Comprobación de la correcta eliminación del elemento galería  
Tabla 37: P.0027- Demostración de los detalles de un elemento al pulsar el id



# Capítulo 1

## Introducción y objetivos

### *1.1 Introducción*

Este proyecto surge en respuesta a la demanda de la empresa Aptent Soluciones S.L.<sup>1</sup>, que solicitó el desarrollo de una aplicación web para la gestión de contenidos web en el marco de un Proyecto Fin de Grado (PFG). La empresa Aptent Soluciones S.L es una spin-off nacida de la Universidad Carlos III de Madrid.

La empresa Aptent es experta en ofrecer diferentes servicios en accesibilidad apoyándose en la innovación y las nuevas tecnologías. Dentro de las iniciativas de Aptent está el proyecto Teatro accesible<sup>2</sup>, cuyo objetivo es crear, fomentar y recoger toda la oferta de teatro accesible para personas con discapacidad sensorial. Para proporcionar ese objetivo, se tiene una aplicación web que recoge la oferta de los teatros que participan en esta iniciativa junto con la información más relevante. Debido al crecimiento y actividad constante de este proyecto, esta aplicación web inicial que gestiona los contenidos se ha quedado insuficiente y ha sido necesario completarla con más funcionalidad y tecnología. Así, nace la idea del proyecto, como el diseño, desarrollo y administración de un gestor de contenidos web a través una aplicación web nueva que sea la ampliación de la actual. Con esta ampliación se quiere asegurar una gestión usable con diferente y más amplio contenido, además de proporcionar un entorno de administración.

El escenario de partida se basa en una tecnología de actualización de contenidos que se ofrecía en una aplicación web a través de un sistema gestor de la base de datos (SGBD), es decir a través de la interfaz del SGBD, lo cual convertía en una tarea muy tediosa para el usuario sin

---

<sup>1</sup> Aptent Soluciones S.L, <http://aptent.es/>

<sup>2</sup> Teatro accesible <http://www.teatrosaccesibles.com/>

conocimiento técnicos el simple hecho de actualizar una noticia o mostrar una galería de fotos. Para solucionar esta situación surge la idea de este proyecto. Con el desarrollo del proyecto se quiere dar soporte de administración a la plataforma para controlar así la gestión de contenidos permitiendo una actualización más ágil y sencilla, asistiendo al usuario en el proceso de publicar o actualizar nuevos contenidos. De esta manera, cualquier usuario, sin necesidad de que tenga conocimientos informáticos puede ser editor y administrador del sistema.

## **1.2 Objetivos**

El objetivo principal de este proyecto es el desarrollo de un sistema de Gestión de Contenidos Web a través de una aplicación web que tiene como funcionalidad añadida un entorno de administración. Este proyecto sirve como complemento a una aplicación web existente. En base a ese objetivo principal, se proponen los siguientes objetivos intermedios:

- Proporcionar un sistema de gestión de contenidos (*Content Management System*, abreviado CMS) a través de una aplicación web, para que el usuario pueda actualizar los contenidos que desee.
- Proporcionar una política de administración en el CMS mediante la creación de un perfil de administrador con privilegios para poder llevar a cabo tareas de administración.
- Dotar al CMS de un manual de usuario para facilitar su utilización de manera que cualquier usuario pueda usar el gestor sin dificultad.

## **1.3 Fases del desarrollo**

Siguiendo un enfoque metodológico, para el desarrollo de la propuesta se han llevado a cabo las siguientes fases: análisis, diseño, implementación de los componentes de software necesarios para extender la estructura actual del gestor de contenidos, así como pruebas para comprobar el correcto funcionamiento.

## **1.4 Medios empleados**

El entorno de trabajo en la empresa Aptent ha facilitado la creación del sistema objetivo de este proyecto y ha marcado diferencia en tiempos de ejecución del proyecto. Aptent es una empresa que se encuentra en el vivero de empresas de la Universidad Carlos III en Parque Científico Universidad Carlos III de Madrid.

El proyecto se ha desarrollado dentro de sus oficinas con la ayuda de sus directores Diego Carrero y Javier Jiménez. Se facilitaron medios software y hardware por esta empresa para el desarrollo del proyecto.

## **1.5 Estructura de la memoria**

La memoria de este proyecto está organizada en nueve capítulos. En este Capítulo se presenta la introducción y motivación del proyecto exponiendo los objetivos que se pretenden alcanzar.

A continuación se presenta una breve descripción de cada uno:

- Capítulo 2: se ofrecen los antecedentes del proyecto así como un sencillo estado de la cuestión, terminando con una discusión de los datos donde se presenta una propuesta preliminar de solución según los objetivos marcados.

- Capítulo 3: se presenta un estudio tecnológico incluyendo información sobre las herramientas y software utilizado.

Los capítulos 4, 5 y 6 incluyen el desarrollo de la propuesta de solución:

- Capítulo 4: se describe la fase de Análisis del proyecto. Se presentan las restricciones de las que se partía al iniciar el proyecto además de los requisitos exigidos y alcanzados para conseguir una solución óptima según los objetivos definidos.
- Capítulo 5: se presenta la fase de Diseño e Implementación donde se describe la arquitectura del proyecto, la interfaz principal y sus características.
- Capítulo 6: recoge la fase de pruebas.

Por último, se presentan:

- Capítulo 7: Manual de usuario para la correcta utilización de la aplicación web objetivo de este proyecto.
- Capítulo 8: Gestión del proyecto: planificación y presupuesto.
- Capítulo 9: Conclusiones y las líneas futuras de trabajo.

# Capítulo 2

## Estado del Arte

### *2.1 Introducción*

En este capítulo se incluyen las bases teóricas fundamentales para el desarrollo del proyecto. Está compuesto por diferentes apartados que permitirán conocer mejor las motivaciones del proyecto que se presenta.

También analiza la tendencia actual que las tecnologías web están llevando como modelos y tipos de aplicaciones que se pueden desarrollar y cómo es su arquitectura.

Finalmente, se realiza un análisis básico de los gestores de contenido indicando cuales son los más utilizados por empresas que compiten en el mismo mercado que el de la empresa Aptent donde se ha desarrollado este proyecto.

Como se puede observar, cada vez es más necesario adaptar el software y los sistemas on-line a la lógica de negocio de cada cliente. Por este motivo, no sólo es necesario que la empresa disponga de un sitio web corporativo, sino que es muy aconsejable una integración total con sus sistemas de contenidos web para obtener un sistema eficiente en todas las áreas de su corporación [1].

El diseño y desarrollo de aplicaciones web consiste en implementar las necesidades, objetivos, ideas de una persona o empresa en Internet utilizando las tecnologías más idóneas según el tipo de proyecto.

En este proyecto, el CMS de la aplicación web cubre las necesidades que la empresa exige y da una clara ventaja al usuario que lo utiliza facilitándole la misión de actualizar los contenidos debido a que solo necesita un navegador para realizarlo.

## 2.2 Antecedentes del proyecto

### 2.2.1 Dominio

En la actualidad las personas con deficiencias auditivas o visuales no tienen acceso a las adaptaciones de contenidos audiovisuales (Subtitulado y Audiodescripción) de forma habitual, salvo por los cauces, llamémoslos, tradicionales (TV, DVD Home, etc.) y aún así todavía hay mucho por hacer. En el DVD Home, Blu-ray Home o en la visualización de contenidos por Internet las adaptaciones de contenidos son inexistentes o casi testimoniales.

Esta situación impide que las personas con discapacidad participen plena y efectivamente en la sociedad, en igualdad de condiciones con las demás. Por eso, debemos romper las barreras que nos impiden tener una actitud accesible y darles un trato equitativo

Las posibilidades para acceder al Cine, la Televisión, Teatro, Vídeo Home o cualquier evento en directo son las siguientes:

- Las personas sordas pueden acceder por medio del subtitulado para sordos y la lengua de signos.
- Para las personas ciegas, las posibilidades con las que se cuentan son las audiodescripciones para ciegos.
- El caso de las personas mayores, nunca contemplados hasta ahora, implica la utilización de un tipo de subtitulado conceptual que viene determinado por las deficiencias en este sector de la población [4].

El uso de tecnologías audiovisuales para la gestión de la accesibilidad hace más eficiente y rápido el proceso de adaptación. Y es que la solución a muchas de las barreras existentes en los sectores de la discapacidad con barreras de accesibilidad no es una cuestión de viabilidad tecnológica, sino más bien de viabilidad económica o de concienciación social.

Para ofrecer soluciones accesibles en el acceso a la cultura, nace la empresa Aptent Soluciones S.L. que ofrece servicios y soluciones tecnológicas con el objetivo de eliminar las barreras de comunicación y de acceso a contenidos audiovisuales, que hacen que las personas con discapacidad sensorial no puedan ejercer su legítimo derecho de acceder, en igualdad de condiciones, a la educación, el ocio y la cultura.

Esta empresa administra dos aplicaciones web:

- <http://aptent.es> : dedicada a ofrecer información sobre la empresa, su misión y los servicios que ofrece.
- <http://www.teatrosaccesibles.com> : Teatro Accesible es una iniciativa pionera cuyo objetivo es crear, fomentar y recoger toda la oferta de teatro accesible para personas con discapacidad sensorial.

Teatro Accesible es un programa integral para dotar de accesibilidad los espacios culturales cuyo objetivo es la supresión de las barreras de comunicación existentes mediante diferentes soluciones que se muestran en la Figura 2.1 y que se describen a continuación:



Figura 2.1: Iconos de accesibilidad (Fuente: empresa Aptent)

1. Subtitulado para sordos: las personas con discapacidad auditiva pueden apoyarse en el subtitulado para seguir la obra.

2. Audiodescripción: a través de unos receptores de audio individuales, la persona con discapacidad visual puede escuchar una voz en off que describe el contenido visual relevante de la obra de teatro.
3. Bucle de inducción magnética: los usuarios de prótesis auditivas (audífono o implante coclear) pueden escuchar el sonido de la obra de teatro de una forma nítida mediante la utilización de un sistema de bucle individual.
4. Sonido de sala amplificado: los usuarios que tengan problemas de audición, o simplemente quieran oír mejor la obra, pueden escuchar el sonido con unos auriculares.
5. Lengua de signos: obras de teatro o musicales pueden ser interpretadas de una manera integrada mediante la lengua de signos.
6. Programas de mano en braille: el hecho de disponer de programas de mano en Braille o cualquier otra información que se desee comunicar es muy útil para las personas ciegas.

Toda la oferta de teatro, así como información relativa a los puntos anteriores se recoge en la aplicación web "<http://www.teatrosaccesibles.com>". A través de esta plataforma, se busca acercar la cultura a la personas con discapacidad sensorial, facilitar el ocio compartido y ofrecer una programación continuada en lugar de un evento accesible aislado, que es lo que, hasta el momento se ha ofrecido a este colectivo.

Tal como se ha indicado en el Capítulo 1 de Introducción, esta plataforma se ha quedado insuficiente y no da un soporte óptimo a la empresa dada su evolución y crecimiento. Debido a esto, este proyecto se basa en el diseño y desarrollo de una aplicación que extienda dicha plataforma de acuerdo a nuevos requisitos,. Se quiere proporcionar un gestor de contenidos que proporcione mediante un entorno de administración una funcionalidad y capacidad más óptima que la que hasta ahora se ha ofrecido.

### ***2.2.2 Aspectos socioeconómicos***

La discapacidad o incapacidad es aquella condición bajo la cual ciertas personas presentan deficiencias físicas, mentales, intelectuales o sensoriales a largo plazo que, al interactuar con diversas barreras, puedan impedir su participación plena y efectiva en la sociedad, y en igualdad de condiciones con las demás [43].

Al hablar del término discapacidad no nos referimos a las deficiencias físicas, mentales, sensoriales o intelectuales permanentes que vive una de cada diez personas en el mundo, en realidad, nos referimos a las barreras físicas (falta de rampas, invasión de espacios entre otros) y de actitud (lástima, indiferencia entre otros) que la sociedad les impone.

El uso de tecnologías audiovisuales para la gestión de la accesibilidad hace más eficiente y rápido el proceso de adaptación. Y es que la solución a muchas de las barreras existentes en los sectores de la accesibilidad y discapacidad no es una cuestión de viabilidad tecnológica, sino más bien de viabilidad económica.

Dada la amplitud y heterogeneidad del mercado a estudiar, se elaboran grupos más pequeños e internamente homogéneos entre sí para poder escoger medidas de actuación más precisas. Las agrupaciones propuestas son las siguientes:

- 1 Personas con discapacidad permanente: el número de personas con discapacidad en España, según datos del INE (1999), es de 3.528.221, lo que supone un 8,8% de la población, y con atención especial al sexo femenino y a la población de 65 a 85 años,



por su superioridad. Dada la heterogeneidad de este grupo, distinguimos una serie de subdivisiones marcadas por características comunes entre ellas, en las que se encontrarían personas con discapacidades:

- Física o motora
  - Sensorial
  - Cognitiva o cerebral
  - Otros: visceral y patológica
  - Discapacidades múltiples o plurideficiencias
- 2 Personas afectadas por la edad: donde se distinguen dos grupos que tienen necesidades específicas por la propia fisonomía de su cuerpo, estos son:
- Personas mayores
  - Niños

Es obvio que existe un mercado latente en el que las tecnologías audiovisuales aun no se han desarrollado de manera amplia y competente para estos grupos de personas. Sólo tecnologías como los dispositivos móviles poseen algún tipo de accesibilidad hoy en día y son pocas empresas las que se encargan de acercar el ocio a los discapacitados.

La accesibilidad audiovisual y empresas como Aptent, o proyectos como Teatro Accesible conllevan una activación del mercado porque genera nuevas oportunidades de negocio de proporcionar servicios de accesibilidad para que las personas con discapacidad puedan acceder al ocio sin ningún tipo de barrera.

Por tanto, aunque empresas dedicadas a la accesibilidad no estén en auge, se tiene actualmente un mercado sin explorar donde cualquier empresa que se atreva tiene un largo camino que recorrer pero sin duda un gran número de clientes que aún esperan unos servicios de ocio personalizados que no se están ofertando. Es decir, lo que cualquier empresa desearía, poca competencia empresarial y gran demanda de producto.

### ***2.2.3 Tecnología de partida***

En este apartado se indican la tecnología de punto de partida del proyecto, pero se explica en detalle en el Capítulo 4 del presente documento.

Todas las tecnologías involucradas en el CMS están en su mayoría impuestas por la empresa y requisitos, siguiendo un modelo Cliente/Servidor para su implementación.

La solución tecnológica que se presenta para el CMS tiene como lenguaje de partida PHP, que es un lenguaje multiplataforma completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Con esta tecnología tenemos capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, un punto a destacar es su gran conectividad con el Gestor de Base de Datos MySQL.

Además, para el desarrollo del CMS se utilizará Yii, un framework orientado a objetos, software libre y de alto rendimiento.

## ***2.3 Tecnologías de desarrollo de aplicaciones web***

En este apartado se va a introducir la terminología para referirse a las aplicaciones web, tipo de arquitectura, así como clasificaciones de distinta tipología donde ubicar una aplicación web. Al fin y al cabo, el CMS que se pretende desarrollar con este proyecto no es más que una aplicación web que da servicio al usuario que la utiliza.

Como modelo de aplicación web el más frecuente es la arquitectura Cliente/Servidor [3] que se muestra en la Figura 2.2. Se define como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. Clientes y servidores son entidades físicas diferentes que operan en conjunto a través de una red para realizar una tarea.

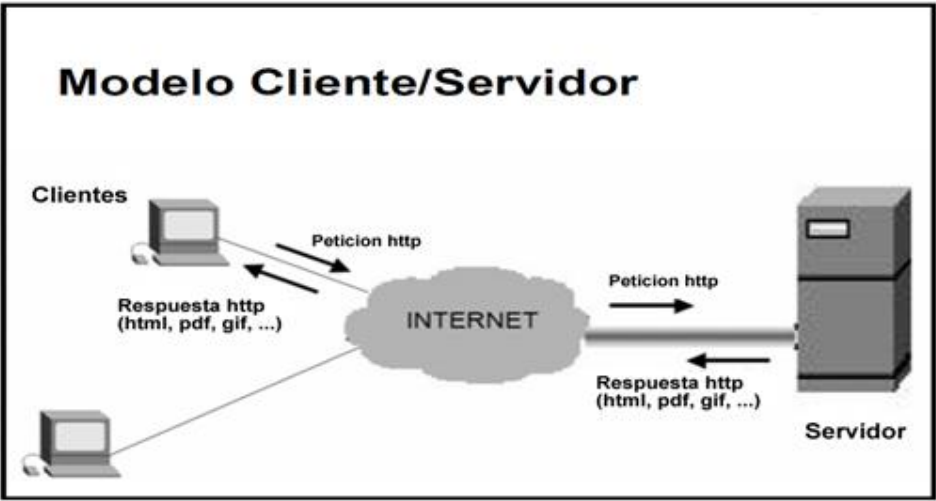


Figura 2.2: Modelo cliente servidor.

La arquitectura Cliente/Servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o dos capas.

- Algunas redes disponen de tres tipos de nodos tal como se muestra en la Figura 2.3: Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.

Servidores de la base de datos que almacenan los datos para los servidores de aplicación. Esta configuración se llama una arquitectura de tres-capas.

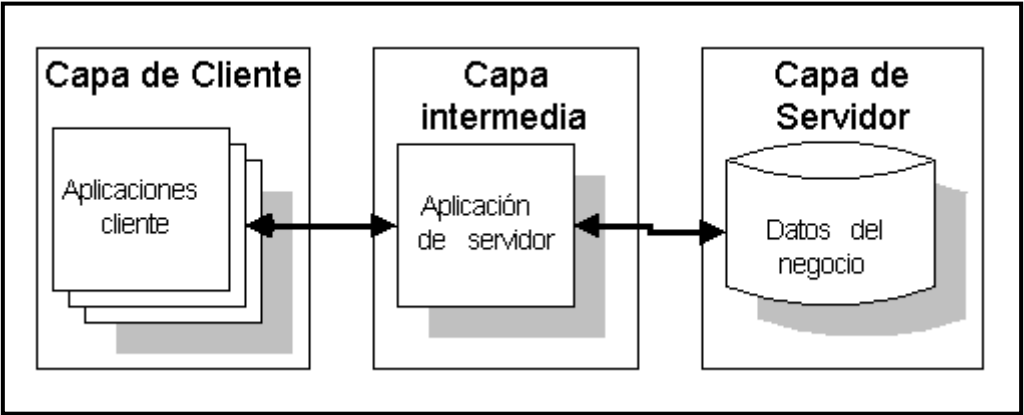


Figura 2.3: Arquitectura de tres capas

La ventaja fundamental de una arquitectura tres-capas (o n-capas) comparado con una arquitectura de dos niveles (o una n-capas con una de dos niveles) es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, otra ventaja primordial de distribuir el trabajo de creación de una aplicación en niveles, es que cada grupo de trabajo se encuentra totalmente separado y aislado del resto de niveles.

En los diseños de aplicaciones actuales, se suelen utilizar las arquitecturas multinivel, y esto es porque a cada nivel, o capa como es conocido comúnmente, se le confía una misión simple, lo que permite que sea de alguna forma, escalable. Esto último quiere decir que, en caso de que las necesidades lo ameriten, pueden ampliarse con facilidad [2].

Toda esta separación de niveles trae como ventaja que el lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas web con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. Inversamente, el lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo [36].

Del lado del cliente tenemos lenguajes como HTML, JavaScript, VBScript, CSS o Flash mientras que del lado del servidor existen lenguajes como CGI, Perl, PHP o XML

Las alternativas para seleccionar una tecnología para el desarrollo de aplicaciones web son variadas; soluciones simples tales como lenguajes tipo Perl, PHP, etc. que pueden solucionar el problema satisfactoriamente pero quizás alguna pueda implicar un coste adicional, y en el transcurrir del tiempo se verán afectadas (como ejemplo Perl tiene la lógica de presentación y la lógica de negocio en un solo archivo).

Para crear una aplicación web con estas tecnologías se necesita tener formación técnica, de ahí el crecimiento de los gestores de contenidos que permiten crear y mantener sitios web a usuarios no técnicos. Con las aplicaciones dinámicas y la Web 2.0 surgen diferentes tecnologías como los framework de tipo front-end que intentan solucionar y hacer este tipo de aplicaciones más flexibles, dotándolas de características dinámicas, que claramente ayudan tanto al usuario como al desarrollador.

A simple vista dentro del modelo de arquitectura Cliente/Servidor podemos encontrar dos tipos de aplicaciones web clasificándolas de manera muy general: las que siempre contienen la misma información y aquellas que se van actualizando a un menor o mayor ritmo de manera constante. La diferencia de los contenidos mostrados nos indicará rápidamente el tipo de página y los recursos que se están utilizando.

- Páginas estáticas: muestran una información invariable y son un recurso rápido y sencillo para tener presencia en Internet. Utilizar páginas estáticas no tiene ningún inconveniente siempre que no se vayan a actualizar los contenidos de manera más o menos frecuente y se tenga que editar el código de cada una de las páginas.
- Páginas dinámicas: las páginas dinámicas pueden alimentarse con datos presentes en bases de datos, el sistema de archivos, programas, etc. El objetivo de usar páginas dinámicas es poder actualizar de manera rápida y sencilla los contenidos, para ello, entre el soporte de datos (normalmente una base de datos) y la presentación de la página (que sigue la misma estructura que las páginas estáticas) se interpone una capa de programación que actúa como filtro y gestor de los datos.

## **2.4 Categorías de Aplicaciones web**

La clasificación de aplicaciones web se puede hacer por varios conceptos, por ejemplo:

- Tipo de contenido (deportes, música, ...)
- Tecnología usada (Flash, HTML, JavaScript...)
- Según el tráfico, país,...

Sin embargo, a efectos de diseño es más interesante clasificarlas según el público al cual va dirigido (target) y que objetivo queremos conseguir con ellas.

Varios autores [9] han definido taxonomías para tratar de clasificar los distintos tipos de aplicaciones web. La más conocida y utilizada en el ámbito de la ingeniería es la de Ginige y Murugesan [10] definiendo las siguientes categorías de aplicaciones web que podemos ver de manera resumida en la Figura 2.4:

- Información (noticias en línea, servicios de noticias, catálogos, manuales),
- Interactivas (formularios de registro, servicios en línea), transaccionales (compras electrónicas, bancos en línea),
- Sistemas de workflow (planificación en línea, administración de inventarios, monitorización de estados),
- Ambientes de trabajo colaborativo (sistemas distribuidos de autoría, herramientas colaborativas),
- Comunidades en línea o redes sociales (grupos de chat, sistemas de recomendación) y portales web (centros comerciales electrónicos, intermediarios en línea).



Figura 2.4: Clasificación de aplicaciones.

Y es que atendiendo a qué tipo de interacción, y tecnología, las aplicaciones web pueden ser clasificadas en la Web 1.0 o en la Web 2.0. Sin un consenso claro está también la Web 3.0 referida a la Web semántica, y existen ya movimientos que caracterizan lo que será la Web 4.0 [5].

Los sitios web de la Web 1.0 entran dentro de la clasificación de páginas estáticas y tienen entre sus principales características:

- Contenidos y sitios estáticos.
- Con fines generalmente comerciales.
- La única función es difundir información
- Con contenidos de alta y baja calidad administrados por un webmaster.

Esto evolucionó hacia aplicaciones web dinámicas gestionadas habitualmente por los sistemas de gestión de contenidos que “servían” páginas HTML dinámicas creadas a partir de realizar operaciones y recuperar información de la base de datos de la aplicación. Es este tipo de web la que se ha elegido como cimiento de la aplicación web que motiva este proyecto.

Entre las características principales de la Web 2.0 destacan [6]:

- Contenidos y sitios flexibles, en permanente transformación por la participación activa del usuario.
- Con fines diversos.
- Permite producir, diseñar, construir y compartir información en diferentes soportes.
- Colaboración: Amplia diversidad en contenidos administrados por usuarios: los usuarios contribuyen a la actualización de los sitios.

Se puede observar en la Figura 2.5 la evolución de la tecnología web con las consecuencias que esto produjo. Entre estas consecuencias destacan el aumento del número de usuarios al que tenía que dar servicio la web con el paso de los años y el incremento de contenidos que estos usuarios lanzaban a la nube. Todo esto provocó el desarrollo de las tecnologías basadas en capas de aplicaciones web antes citadas.

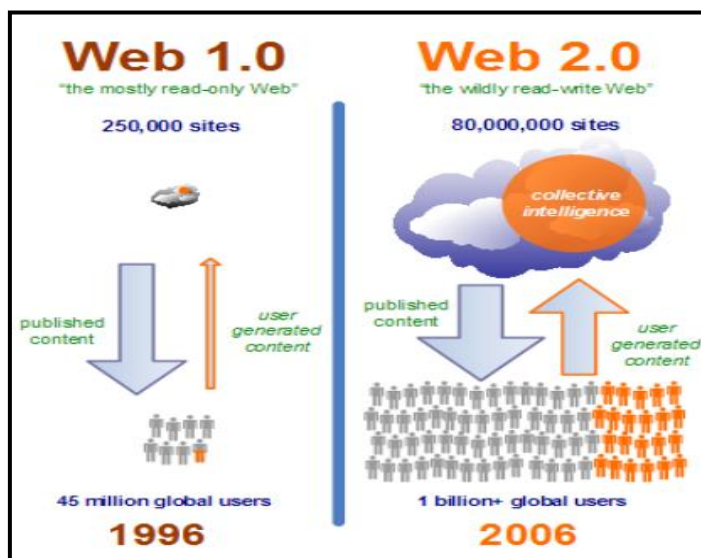


Figura 2.5: Evolución de la Web

La aplicación web en este proyecto trabaja con contenidos y recursos que continuamente cambian y se actualizan por lo que se enmarca dentro de las web dinámicas. Además encaja con las características principales de la Web 2.0 y se centra en el desarrollo de un gestor de contenidos.

A continuación se exponen los diferentes modelos en los que esta aplicación web puede basarse según sus expectativas. La evolución del desarrollo de aplicaciones web [7] tiene, como hemos visto, una trayectoria conocida en este ámbito, por su desarrollo rápido y complejo, esto hace que la estructura de la web evolucione para dar servicios más completos y complejos. A su vez esto provoca que los modelos de aplicaciones web evolucionen a la par. Podemos resumir los principales en los siguientes puntos:

- Modelo 1: Son las más primitivas. Se identifican con este modelo las clásicas aplicaciones web CGI, basadas en la ejecución de procesos externos al servidor web, cuya salida por pantalla era el HTML que el navegador recibía en respuesta a su petición. Este modelo se representa según la Figura 2.6.

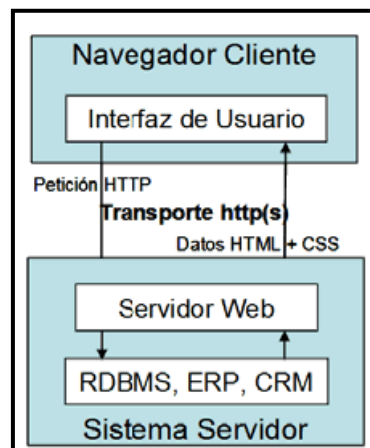


Figura 2.6: Modelo clásico de aplicación web

- Modelo 2: Como evolución del modelo 1 surge el patrón Modelo-Vista-Controlador (MVC) en este tipo de aplicaciones, representado en la Figura 2.7. Se aprecia la incorporación de un elemento controlador de la navegación de la aplicación. Es el patrón de diseño que ha llegado a dominar el espacio de programación de aplicaciones web.

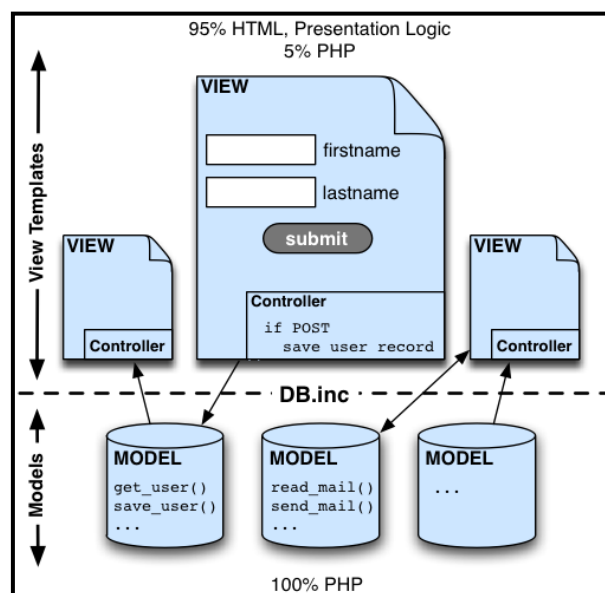


Figura 2.7: Modelo MVC de una aplicación web

Este modelo de arquitectura de software separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: modelo, vista y controlador... Todo lo referente a este modelo y su relación con el CMS propuesta será detallado en el Capítulo 4 de Análisis.

Los entornos de programación (frameworks) que siguen un MVC han pasado a ser una parte importante de cualquier proyecto web. Hoy en día, la evolución de la propia Web y de los estándares, han sido el caldo de cultivo de la aparición de un gran número de frameworks para facilitar y estandarizar la programación web.

Hay una amplia gama de frameworks de desarrollo de aplicaciones web disponibles para Linux que son distribuidos bajo licencia Open Source lo que permite su utilización en todos los

proyectos independientemente de la infraestructura de despliegue. Actualmente existe una inmensa variedad y pueden elegirse según unos requisitos muy específicos, dependiendo de lo que la aplicación web necesite.

Las soluciones para el desarrollo llevado a cabo en este proyecto han sido utilizar para la aplicación web el modelo MVC y la utilización de PHP como lenguaje de programación del lado del servidor ya que es gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

Además se utilizan tecnologías como MySQL y Yii Framework con características muy específicas que aumentan el rendimiento de la aplicación web que se va a desarrollar. Toda esta información se amplía en el Capítulo 4.

## 2.5 *Gestor de contenidos*

Actualmente el mercado de CMS es muy amplio y variado, muchos de ellos son gratuitos [8]. Para las empresas lo interesante no es coger un tipo de CMS muy específico, sino elegir uno que pueda conjugar varias funciones: publicación de noticias, foros, galerías de fotos, reproductores multimedia,... Muchos trabajan con esta diversidad de funciones.

Los CMS se pueden clasificar según sus características:

- Según el lenguaje de programación empleado, como por ejemplo: Active Server Pages, Java, PHP, ASP.NET, Ruby On Rails, Python, PERL
- Según la licencia: Código abierto o Software propietario

Entre los principales gestores del mercado actual se han seleccionado los tres más usados basándome en opiniones personales que provienen del propio uso de estos CMS y del estudio que presentó en 2012, el Centro de Apoyo Tecnológico a Emprendedores de la Fundación Parque Científico y Tecnológico de Albacete: “Estudio de los sistemas de gestión de contenidos web. Análisis de las mejores soluciones del mercado” [12]:

- Desarrollo web en WordPress<sup>3</sup>

El rey de los CMS en España. Es el más rápido, como de trabajar y gestionar, además de que cada vez permite más funcionalidades como implementar una tienda online de forma mucho más simple que con los demás CMS.

Con cualquier proyecto semilla es una muy buena opción para hacer el test de mercado y poder migrar en el momento de dar el salto a una aplicación web compleja.



- Desarrollo web en Drupal<sup>4</sup>

Drupal sería el contrapuesto de WordPress. Es la mejor alternativa desde el punto de vista de rendimiento, escalabilidad, seguridad y estabilidad, pero en contrapunto, es el más complejo de los tres que se están exponiendo, lo que supone que un desarrollo en Drupal sea más largo y costoso y a la vez hace que el mantenimiento por parte de usuarios no técnicos sea muy difícil y limitado.



---

<sup>3</sup> WordPress, <http://es.wordpress.com/>

<sup>4</sup> Drupal, <http://drupal.org.es/>



Si se quiere llevar a cabo un proyecto de alta complejidad, ya sea por proyectos o por número de usuarios, Drupal será una buena elección a priori.

- Desarrollo con Joomla<sup>5</sup>

Joomla ofrece una buena relación entre complejidad y resultados. Dispone de una interfaz de administración, que a pesar de no ser tan simple como la de WordPress, sí que es lo suficientemente sencilla para ser comprendida y utilizada por parte de personas con pocos conocimientos técnicos.



En algunas versiones no estándar nos permite abordar proyectos mucho más complejos y completos que WordPress, y ofrece también innumerables extensiones disponibles para ampliar su funcionalidad original, aunque integrar estos componentes es un poco más complicado que con WordPress.

Gracias a su gran comunidad de usuarios es mucho más fácil detectar fallos de seguridad en el código y permite también realizar sitios web con capacidad de escalada y bastante estables.

Como se puede observar cada uno de los CMS vistos tiene sus variantes tanto positivas como negativas. Sin embargo, no hay uno mejor que otro, puesto que la elección definitiva se realiza en función de los requisitos que se precisen para cada proyecto.

A grandes rasgos, Wordpress se utilizaría para proyectos de portales web básicos, sin demasiada complicación en cuanto a gestión. Joomla, por su parte, permite la creación de sitios o aplicaciones web más complejas aunque precisa de unos conocimientos algo más técnicos por parte del usuario. Finalmente, Drupal sería la elección más completa pero, de igual manera, sería la más compleja de utilizar por parte del usuario, que necesitaría la ayuda de un técnico para su desarrollo y futuras modificaciones [9].

Con este proyecto lo que se pretende es saltar los muros que supone ceñirse a un tipo de CMS creando uno propio. Este gestor tendrá muchas de las características de los CMS más populares pero estará totalmente personalizado para el sitio web que administra lo cual dará una funcionalidad mucho más específica y eficiente.

Esta decisión permite trabajar de manera muy distinta a los gestores citados anteriormente, permite tener un modo de administración totalmente personalizado consiguiendo que sin conocimientos de programación cualquier usuario pueda añadir contenido en el sitio web.

## 2.6 Trabajos relativos

En este apartado se describen empresas del ámbito de Aptent que se dedican a eliminar las barreras de comunicación y de acceso a servicios de contenidos audiovisuales. Se analiza a grandes rasgos cómo gestionan y administran sus sitios web.

Existen actualmente en el mercado múltiples empresas que ofrecen un servicio parecido al que Aptent y Teatro Accesible desarrollan. Cada vez se es más consciente de la necesidad de llegar a todos los segmentos de la sociedad, incluyendo a aquellos que necesitan un acceso específico para poder tener las mismas oportunidades de acceso a la cultura y ocio.

- “Edsol”<sup>6</sup> es una empresa productora de contenidos audiovisuales, incorporan para sus producciones subtítulos, lenguaje de signos y bucle magnético en auditorios, aulas y

---

<sup>5</sup> Joomla, <http://www.joomla.org/>

<sup>6</sup> Edsol, <http://www.edsolproducciones.com/>



salones de actos. En cuanto a la oferta de contenido web no tiene un gran volumen ya que su principal misión es ofrecer sus servicios a empresas externas y no al público. Su gestor de contenidos en referencia a su sitio web no tiene una complejidad muy grande y el contenido que maneja no tiene apenas volumen.

- “Signia”<sup>7</sup> es una asociación sin ánimo de lucro que presta servicios de interpretación a personas sordas y guía-interpretación a personas sordo-ciegas.
- “36 caracteres”<sup>8</sup> es una empresa de traducción audiovisual especializada en subtitulado de cine y televisión, sobretitulación de teatro y ópera y otros servicios de accesibilidad audiovisual, traducción, posproducción y gestión asociados a estas labores.
- 36 Caracteres o Signia manejan quizás un volumen de contenidos mucho más amplio. Al igual que la aplicación de Teatro Accesible estos contenidos deben estar administrados y controlados mediante un CMS.

En este caso Signia o 36 Caracteres utilizan el CMS WordPress como gestor para el desarrollo de su página web. Como se ha analizado antes, la elección de este gestor dependerá siempre del tipo de proyecto que se realice.

## ***2.7 Discusión de los datos y propuesta de solución***

En este capítulo se ha presentado el dominio de aplicación del proyecto, y una breve descripción tecnológica del proyecto para pasar a ver qué tecnologías en el desarrollo de aplicaciones web hay disponibles en la actualidad, así como trabajos relativos.

Este capítulo se ha introducido ciertas tecnologías que se han ido citando a lo largo de los apartados anteriores. Se han mostrado propuestas tecnológicas pero no se ha podido comprobar que unas tecnologías fuesen más apropiadas que otras. Esto es debido a que el proyecto estaba supeditado en cierta manera a las bases de otro (la pagina web de Teatro Accesible). Este sitio web ya tenía su propia arquitectura, lenguaje de programación y base de datos.

Si bien es cierto que se volvieron a analizar las características del proyecto antes de su comienzo, se valoró que seguía siendo más optimo utilizar el modelo de arquitectura MVC para la creación del CMS que estará basado en PHP y MySQL y que utilizará la tecnología de Yii Framework para aumentar su rendimiento.

---

<sup>7</sup> Signia, <http://www.asociacionsignia.org/>

<sup>8</sup> 36 caracteres, <http://www.36caracteres.net/>

# Capítulo 3

## Estudio tecnológico

### **3.1 Introducción**

En este capítulo se presenta un estudio de las tecnologías y de su selección para implementar la propuesta de solución del proyecto, una aplicación web con entorno de administración para la gestión de contenidos del sitio web de Teatro accesible. A continuación se exponen las decisiones que han llevado a la elección del entorno de ejecución, del servidor web y del Sistema Gestor de Bases de Datos.

### **3.2 Herramientas de desarrollo**

Las herramientas utilizadas para el desarrollo de la aplicación web se detallan en este apartado. Estas herramientas fueron elegidas por las características que se citan a continuación aunque su utilización en este proyecto ha sido impuesta desde un principio por preservar la unidad con la página web que este gestor administra.

#### **3.2.1 HTML**



HTML [13] es el estándar y un lenguaje de marcado predominante para la construcción de páginas web. HTML no describe los aspectos de presentación de un documento web sino que ofrece a cada plataforma la información de su contenido y estructura para que le dé formato según su capacidad y la de su navegador (tamaño de la pantalla, fuentes que tiene instaladas, etc.).

Durante la creación de HTML no se tenía conciencia de que la Web llegaría a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar

respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML.

Actualmente (octubre de 2012) el WHATWG prosigue su desarrollo continuo del HTML y el W3C, de acuerdo con su plan de trabajo para 2014 [14] está elaborando la recomendación del HTML 5 (para 2014).

El diseño web, aparte de cumplir con las especificaciones propias del lenguaje HTML, debe respetar ciertos criterios de accesibilidad web, siguiendo unas pautas, normativa y leyes vigentes en los países donde se regule dicho concepto. Dichos criterios están disponibles y han sido desarrollados por el W3C a través del estándar de las Pautas de Accesibilidad al Contenido Web o *Web Content Accessibility Guidelines* (WCAG) 2.0[15]. Las WCAG comprenden pautas que proporcionan soluciones de diseño y que utilizan como ejemplo situaciones comunes en las que el diseño de una página puede producir problemas de acceso a la información a ciertas personas con diferentes tipos y niveles de discapacidad. Las Pautas contienen además una serie de puntos de verificación y técnicas que ayudan a detectar posibles errores de accesibilidad.

### 3.2.2 JavaScript



La evolución anárquica que ha tenido HTML ha supuesto toda una serie de inconvenientes y deficiencias que han debido ser superados con la introducción de otras tecnologías accesorias capaces de optimizar y automatizar el funcionamiento de las aplicaciones web. Un ejemplo de esto es JavaScript entre otros.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript [16]. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y generando páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

Todos los navegadores actualizados interpretan el código JavaScript integrado en las páginas web. Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

El comité ECMA está trabajando en el desarrollo de la próxima versión de Javascript conocida con el nombre de Harmony. En este sentido, si se cumplen los plazos previstos Harmony estaría listo para finales del 2013.

Actualmente Javascript es uno de los lenguajes de programación más visitados del lado del cliente como se muestra en la Figura 3.1.

© W3Techs.com	usage	change since 1 July 2013
1. JavaScript	89.2%	-0.1%
2. Flash	17.7%	-0.6%
3. Silverlight	0.2%	
4. Java	0.1%	
percentages of sites		

Figura 3.1: Estadísticas lenguajes de programación más visitados del lado del cliente [17]

### 3.2.3 PHP



PHP [18] es un lenguaje de script, diseñado para, entre otras cosas, aumentar, incrementar el dinamismo de las páginas web. Originalmente se trataba de un conjunto de macros concebidas para ayudar en el mantenimiento de páginas web. Desde entonces, sus características han ido creciendo hasta convertirse en un lenguaje de programación completo, capaz de manejar entornos que integran grandes bases de datos. Su popularidad se basa, en gran parte, a su sintaxis similar a la del lenguaje de programación C, su rapidez y simplicidad.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Actualmente, PHP es el lenguaje de programación del lado del servidor más usado como se muestra en la Figura 3.2.

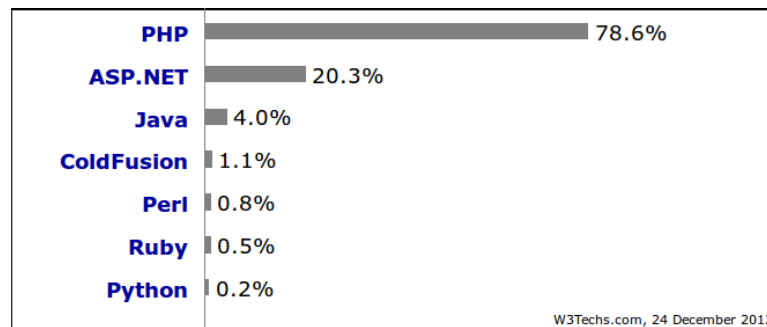


Figura 3.2: Estadísticas de uso de lenguajes de programación de servidores [17]

PHP es también el lenguaje más popular entre los ordenadores que utilizan servidores Apache como servidor web. La versión más reciente de PHP es la 5.4.16 del 6 de Junio de 2013.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP es un desarrollo "Open Source". Es decir, se puede ver y modificar el código fuente de la aplicación siempre y cuando cumplas con su licencia PHP. También es gratuito. Resumiendo las características de PHP, entre sus principales ventajas destacan:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP 5).

En este proyecto se ha utilizado el lenguaje de lado servidor PHP.

### 3.2.4 Sistema Gestor de Base de Datos: MySQL



MySQL [19] es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario. MySQLAB (subsidiaria de Sun Microsystems) desarrolla MySQL como software libre en un esquema de licencia dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privados pueden comprar a la empresa una licencia específica que les permita este uso.

MySQL es muy utilizado en aplicaciones web. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL se ha elegido como sistema gestor de la base de datos de la plataforma en este proyecto por su escalabilidad, conectividad, versatilidad y disponibilidad como software libre. La única limitación en cuanto al tamaño de una base de datos viene determinada por los recursos del sistema. MySQL permite gestionar bases de datos del orden de seis mil tablas y alrededor de cincuenta millones de registros.

### 3.2.5 Yii Framework



Yii es un framework [20] PHP basado en componentes de alto rendimiento para desarrollar aplicaciones web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo.

Yii es un framework genérico de programación web que puede ser utilizado para todo tipo de aplicaciones web. Gracias a que es liviano de ejecutar y está equipado con soluciones de cacheo sofisticadas, es adecuado para desarrollar aplicaciones de gran tráfico como portales, foros, sistemas de administración de contenidos (CMS), sistemas de comercio electrónico (e-commerce), etc.

Como la mayoría de los frameworks PHP, Yii es un framework MVC y sobresale frente a otros por su eficiencia, su gran cantidad de características y su clara documentación. Yii ha sido diseñado cuidadosamente desde el principio para el desarrollo de aplicaciones web. Esto se ve reflejado en la Figura 3.3.

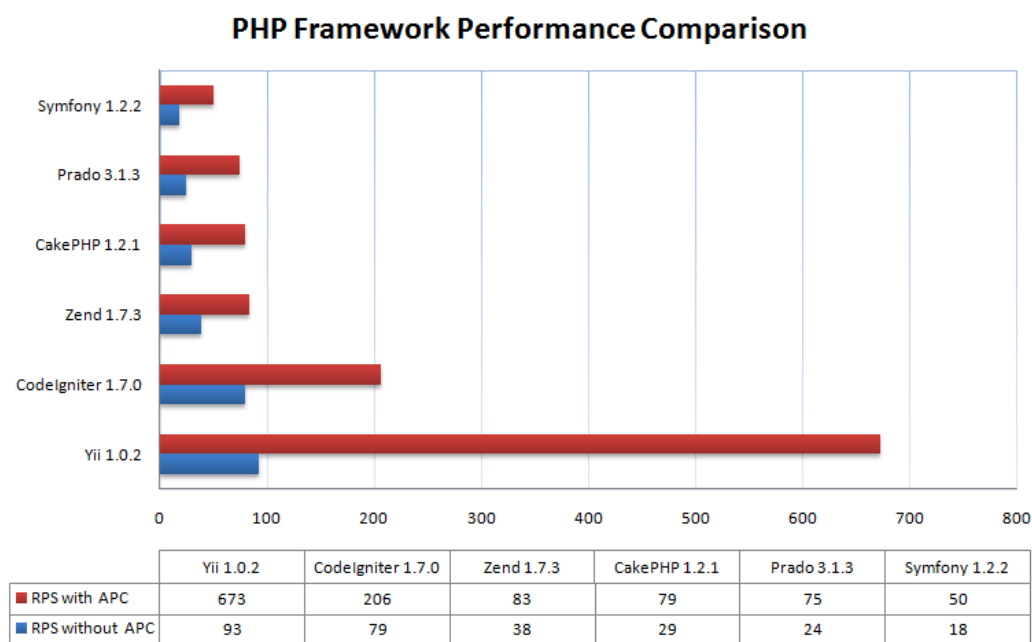


Figura 3.3: Gráfica comparativa de frameworks en PHP [21]

### 3.3 Tecnología entorno de desarrollo

En este apartado se describirán las herramientas utilizadas para el desarrollo del CMS propuesto de este proyecto. Los entornos que a continuación se citan al igual que el resto de la tecnología fueron impuestos o condicionados por la tecnología del origen del proyecto.

#### 3.3.1 APACHE Server



El servidor HTTP Apache [22] es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft [23]).

### 3.3.2 Aptana Studio 3.0



Aptana Studio[24] es un entorno de desarrollo integrado de software libre basado en Eclipse [25] y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: PHP, Python, CSS, Ajax, HTML. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

Entre sus principales características destacan:

- Asistente de código para HTML y Javascript.
- Librerías Ajax (jQuery, prototype, scriptaculous, Ext JS, dojo, YUI y Spry entre otras).
- Conexión vía FTP, SFTP, FTPS y Aptana Cloud.
- Herramientas para trabajo con base de datos.
- Marcado de sintaxis mediante colores.
- Compatible con extensiones para Eclipse.

### 3.3.3 Xampp



XAMPP [26] es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. De hecho su nombre viene de ahí, X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP) y P (Perl). XAMPP es independiente de plataforma y tiene licencia GNU GPL. Existen versiones para Linux, Windows MacOS X y Solaris.

Una de las ventajas de XAMPP [27] es que de una forma muy sencilla y rápida se puede montar un entorno de desarrollo de cualquier aplicación web que use PHP y base de datos.

### 3.3.4 TeamLab



TeamLab [28] es una oficina multifuncional albergada en “la nube”. Permite manejar documentos, proyectos, relaciones de equipo y clientes en un solo lugar.

Además de las opciones necesarias para la gestión de documentos, proyectos, clientes y equipo en el nivel alto, TeamLab proporciona un conjunto de opciones únicas: ofrece una red social que incluye blogs, foros, encuestas, marcadores y eventos.

También se puede ajustar a las necesidades específicas: desde branding corporativo hasta el flujo de ventas personalizado.

### **3.3.5 Google Chrome**



Google Chrome [29] es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto.

Chrome está diseñado para ser rápido en todos los aspectos. Se inicia rápidamente carga las páginas web de forma instantánea y ejecuta aplicaciones web complejas a gran velocidad. Estas características hacen que sea una herramienta de depuración de código muy útil a la hora de testear aplicaciones web y ha sido de gran ayuda a la hora de realizar las pruebas de depuración y comprobación de la aplicación web.



# Capítulo 4

## Análisis

### **4.1 *Introducción***

En este proyecto se ha llevado a cabo un desarrollo siguiendo un enfoque metodológico y en este capítulo se va a presentar la información relativa a la fase de Análisis del proyecto. Se ha hecho la captura de requisitos del entorno objetivo del proyecto.

En primer lugar, se va a introducir la tecnología ya desplegada de partida del proyecto, para comprender y contextualizar algunas decisiones en el Análisis del sistema, además se analizará en profundidad la arquitectura de este sistema que se está desarrollando, así como su posterior Diseño e Implementación, y Validación. Después, se pasa a indicar los requisitos del sistema.

### **4.2 *Requisitos impuestos y arquitectura del sistema***

Es importante tener en cuenta durante todo este Capítulo que hay una tecnología de partida, la plataforma del sitio web del proyecto Teatro Accesible. Esto ha condicionado en gran medida la tecnología y desarrollo del proyecto, puesto que han sido impuestas distintas restricciones al seleccionar las herramientas, la arquitectura y el entorno de trabajo a utilizar.

Para tomar la decisión acerca de qué tipo de arquitectura se debe utilizar en la aplicación a desarrollar se tuvieron en consideración tanto los requisitos funcionales como los no funcionales, los cuales han sido descritos en apartados anteriores, como el hecho de que se trata de una aplicación web, y que los lenguajes que van a ser utilizados son PHP y HTML.

Sin embargo, como se ha citado en capítulos anteriores, la página web de Teatro Accesible ya utiliza un modelo de arquitectura MVC, por lo que puesto que el proyecto que se

desarrolla comparte los cimientos de la aplicación web de Teatro Accesible, deberá utilizar y basarse en la lógica de negocio que utiliza éste.

Los motivos que llevaron a elegir el modelo arquitectónico MVC, son que en primer lugar, es sencillo conceptualmente y en segundo lugar, logra el objetivo de dividir las diferentes partes que conforman una aplicación, con el objeto de mantener el código fuente. Puesto que la aplicación a desarrollar pretende ser escalable, con esta arquitectura existe una clara división del uso del código desarrollado, por lo que son fácilmente identificables las funcionalidades descritas en el código de la aplicación [37].

A partir de la arquitectura MVC se realiza la siguiente división tal como se aprecia en la Figura 4.1:

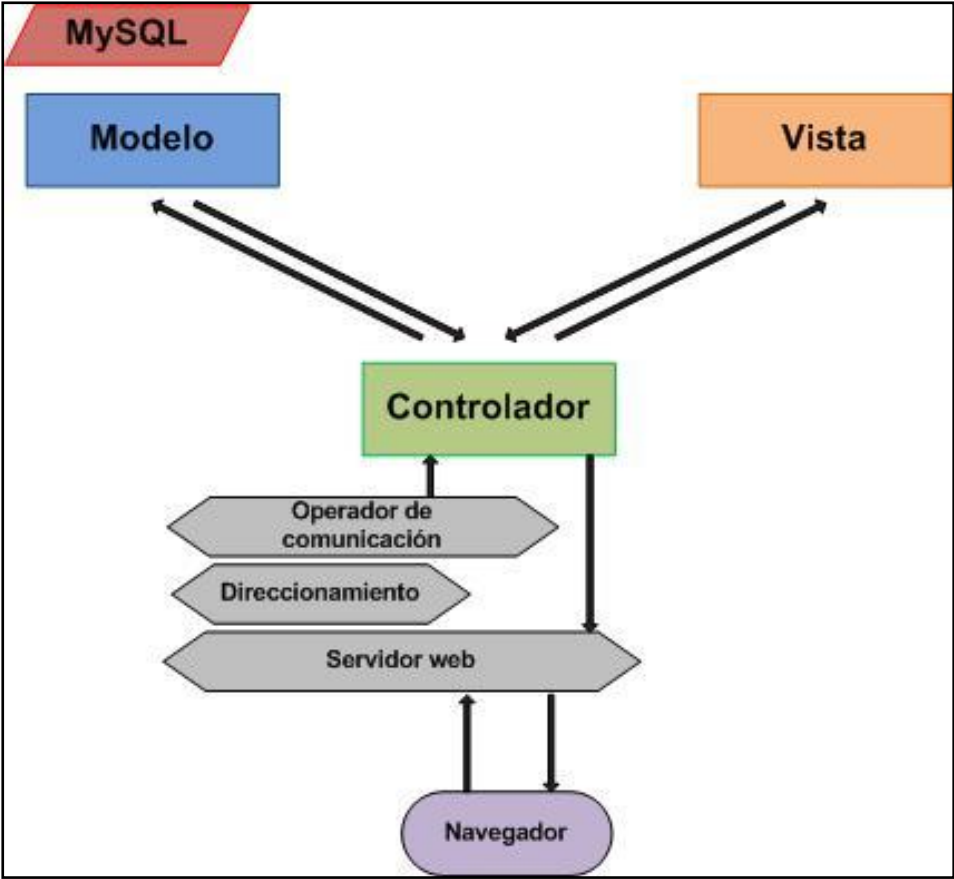


Figura 4.1: Modelo MVC por capas.

Modelo: el modelo tiene entre sus funciones acceder a la capa de almacenamiento de datos. La mejor práctica es manejar una capa de acceso a datos de manera independiente al modelo y aunque no es óptimo, igualmente es válido otorgar a los modelos capacidades de acceso a datos.

1. Define las reglas de negocio (la funcionalidad del sistema).
2. Lleva un registro de las vistas y controladores del sistema.
3. Si es un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

Vista: será el diseño estético de la aplicación. Es la parte gráfica, la parte que el usuario del sistema verá en su navegador web. Esta parte estará formada por las clases clasificadas como “vistas” de la aplicación.

Controlador: Es empleado como un mediador entre el gráfico (la Vista) y el modelo (Modelo), coordinando las acciones que son llevadas a cabo entre ambos. Los controladores son las acciones que realiza la aplicación en respuesta a una petición, son el motor de la aplicación.

Para seguir el modelo MVC se impuso como requisito la utilización de Yii Framework, un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones web de gran escala. Permite la máxima reutilización en la programación web y acelera el proceso de desarrollo.

Como se observa en la Figura 4.2 en la implementación del patrón MVC, Yii introduce un controlador adicional, denominado Application, que permite encapsular el contexto de ejecución para el procesamiento de una petición. Application contiene información sobre la petición del usuario de tal forma que pueda ser gestionada por el controlador apropiado.

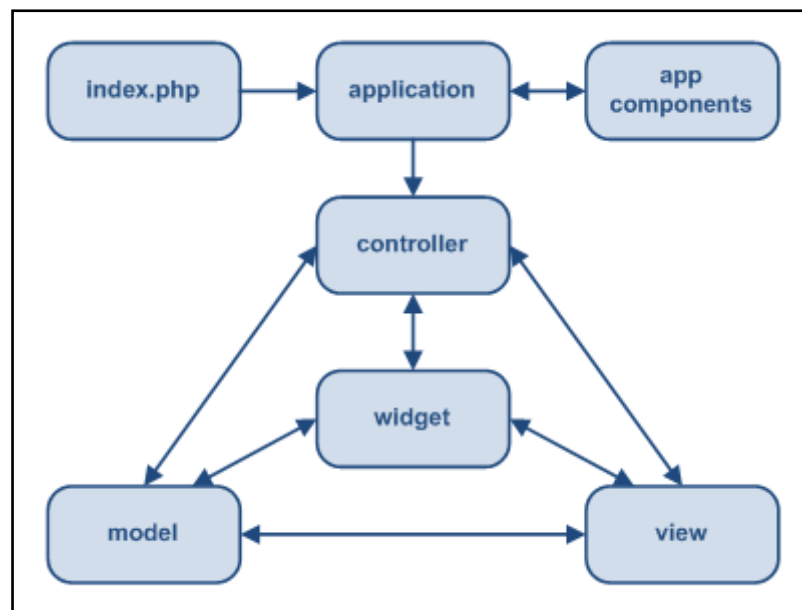


Figura 4.2: Estructura estática de aplicación Yii.

El diagrama de la Figura 4.3 muestra el ciclo de vida de la aplicación durante el procesamiento de una petición de usuario.

1. El usuario realiza una petición a la URL. El servidor web maneja la petición ejecutando el script de arranque `index.php`.
2. El script de arranque crea una instancia de la clase `Application` y la ejecuta.
3. La aplicación obtiene información de la petición del usuario del componente `request`.
4. La aplicación determina el controlador y la acción que manejarán la petición con la ayuda del componente `urlManager`.
5. La aplicación crea una instancia del controlador solicitado para resolver la petición del usuario. Asimismo, el controlador determina la acción solicitada y comprueba si puede ser ejecutada según los filtros asociados. La acción solamente será ejecutada si los filtros lo permiten.
6. La acción recupera algún modelo de la capa de datos.
7. La acción realiza la llamada a la vista correspondiente con el modelo recuperado.

8. La vista lee y muestra los atributos del modelo.
9. Adicionalmente, la vista puede instanciar algún *widget*.
10. El resultado de la vista puede ser incrustado dentro de un *layout*.
11. La acción completa la vista y se la devuelve al usuario.

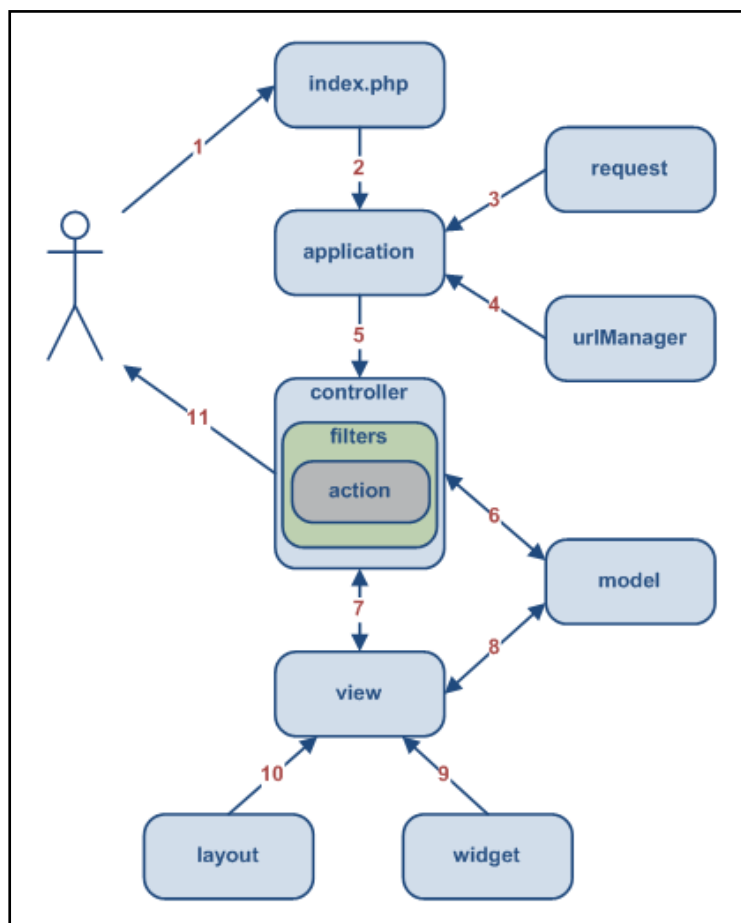


Figura 4.3: Ciclo de vida de una petición

Como ya se ha explicado en apartados anteriores y para generar una visión global del CMS a desarrollar, la nueva aplicación web utiliza como tecnologías web el Yii Framework sobre servidor Apache con PHP como lenguaje de programación de lado servidor y MySQL como gestor de Base de Datos.

Otro de los requisitos que ha sido impuesto por los motivos expuestos anteriormente es el lenguaje de programación PHP, que además ha venido impuesto por el framework utilizado. Es buena alternativa como tecnología de lado servidor ya que proporciona ventajas como que es un lenguaje muy utilizado de programación para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante, por lo que aunque se trate de una restricción, realmente es beneficioso para el proyecto que se va a desarrollar. Otro de los motivos favorables de utilizar PHP es que éste puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún coste [38].

Sobre el alojamiento web, la aplicación web del proyecto Teatro Accesible se encuentra en uno de los hosting más eficientes [30] que actualmente hay en el mercado, Strato. Este hosting permite crear páginas web individuales y profesionales. Los paquetes avanzados dan

soporte a las tecnologías PHP, Perl, MySQL y AppWizard, lo cual proporciona facilidad para la instalación y despliegue de un portal básico vacío con vistas de administración. Incluye servidores configurados específicamente y modificaciones en el servidor web Apache.

Tal como se muestra en la figura 4.4, dentro del servidor Apache se encuentran instalados los paquetes de PHP (instalados por defecto) y el entorno de desarrollo de Yii Framework que proporciona el servicio perfecto para conseguir la arquitectura del gestor de contenidos basado en una aplicación web que proporciona a su vez un entorno de administración para el usuario.

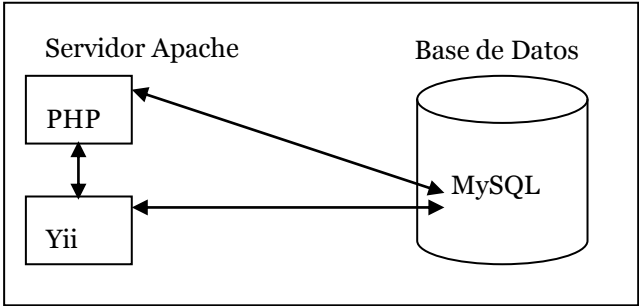


Figura 4.4: Arquitectura del proyecto.

### 4.3 Arquitectura de la Información

#### 4.3.1 Precedentes

Para el poder hacer el análisis del sistema del CMS, es necesario partir de la arquitectura de la información que tiene la plataforma existente del proyecto Teatro Accesible.

El sitio web comprende una serie de unidades de contenido, a las cuales se accede a través de menús (tal como se muestra en la Figura 4.5). Esta información y estructura va a provocar una restricción de la interfaz del CMS así como la implementación de clases y la lógica de diseño. Esto es debido a que los servicios que ofrece esta página se encuentran desplegados en estos menús y todos los contenidos web con los que el CMS va a trabajar se encuentran en ellos.

Es importante conocer las secciones en que se dividen estos menús para la interfaz del CMS de cara a entender qué elementos o contenidos va tener que permitir administrar el gestor propuesto.



Figura 4.5: Opciones del menú de Teatro Accesible.

Para comprender bien la estructura de la información, y por tanto del gestor de contenidos, a continuación se muestran los apartados del mapa web de aplicación web.

#### ❖ Inicio

- Twitter
- Facebook

En su página de inicio podemos ver las obras de teatro más representativas que se están realizando de manera accesible, un menú directo de Twitter donde poder estar alerta de cualquier nueva información referente a obras o teatros e información sobre la empresa promotora, Aptent, patrocinadores... además de poder elegir el idioma principal de la web.

#### ❖ Nosotros

- Soluciones
- Patrocinadores
- Promotor
- Colaboradores
- Asociaciones
- Amigos

En el apartado “nosotros” se explica brevemente la iniciativa de la empresa y se detalla en cada apartado todas las empresas colaboradoras o promotoras que han ayudado a construir este proyecto como ONCE, Fundación Vodafone...

#### ❖ Programación

- Ciudades
- Teatros

En el grupo de Programación se muestra al usuario las obras que actualmente están activas junto con su horario, fecha y teatro. Dentro de este apartado podemos elegir ver la programación actual filtrando por ciudades o teatros si queremos hacer una selección mucho más exhaustiva.

#### ❖ Archivo

- Temporada 2012/13
- Festival de Mérida 2012
- Temporada 2011/12

El apartado Archivo es un histórico donde quedan guardadas todas las obras que se han hecho de manera accesible además de los festivales en los que se ha participado.

#### ❖ Noticias

- 2013
- 2012
- 2011

Como su propio nombre indica este apartado es el de Noticias en constante actualización referentes tanto a nuevas obras que se vayan a realizar como a noticias de interés en el ámbito de la accesibilidad.

#### ❖ Galerías

- Subtítulos
- Temporada 2012-13
- Mercantes 2012

- Festival de Mérida
- Rueda de Prensa

Hay un apartado denominado Galerías, que incluye una galería de fotos y archivos donde el usuario puede ver las imágenes que se han ido tomando en una obra, fotos referentes a ruedas de prensa o festivales en los que Teatro Accesible ha participado.

#### ❖ Contacto

Por último, hay un apartado de contacto que está al servicio del usuario que puede ponerse en contacto con la empresa para resolver dudas, dar opiniones o valorar el trabajo de ésta.

### 4.3.2 Decisión de la arquitectura de información

Teniendo en cuenta la arquitectura de información de Teatro Accesible que se ha desglosado en el apartado anterior, se decidió que para tener una mejor simbiosis con la Web de esta plataforma, que la web de administración, es decir, el CMS que se desarrolla, formara parte de su árbol de páginas web como una “hoja” más.

Con esto se deduce que este CMS no tendrá una página web alternativa sino que la raíz de su propia dirección web estará formada por la web que alberga todo lo referente a la aplicación en el servidor.

“Dirección\_IP\_Servidor:8081/teatroaccesible/site/login” (por seguridad y a petición de la empresa cliente no se revela en la memoria la dirección IP para acceder al servidor que contiene la aplicación web del CMS) será la URL a la que finalmente se tendrá que acceder si se quiere administrar la página web.

Esta decisión orientada a los datos es sencilla y ventajosa ya que se tendrán todos los datos albergados en el mismo servidor web sin problemas de compatibilidades, además esto da facilidad a la hora de acceder para cualquier administrador y ciertamente es una elección mucho más elegante.

Dentro de la base de datos de partida se encuentran distintas tablas (TEATROS, CIUDADES o NOTICIAS) que tienen dentro distintos atributos. Por ejemplo, en el caso de CIUDADES se tienen atributos como Id, Visible y Name, es decir, un campo que indica el número de identificación de la ciudad, la elección de si se desea que esté visible al público o no (con 0 o 1) y un campo para dar nombre a la ciudad que se guarda en la base de datos.

En resumen, y para poder entender de una manera global este proyecto; Teatro Accesible necesita una actualización continua de datos y contenidos en su Web. Esto antes de comenzar este proyecto era algo muy tedioso, la única manera de poder hacerlo era entrar en la base de datos a través de su gestor de administración y rellenar elemento por elemento las tablas que se quisieran actualizar de manera que la página al renderizar de nuevo actualizase los contenidos. Con un ejemplo puede verse el alcance de dicha acción.

Se supone: con motivo de la celebración de San Isidro en Madrid se realizan tres obras de teatro accesible en el teatro Valle-Inclán. Para que los usuarios de esta plataforma tengan un conocimiento continuo de las ofertas ofrecidas por Teatro Accesible es necesario que éste muestre en su web la noticia antes anunciada. Esto implica que se deben añadir datos en muchas de las tablas de la base de datos como Teatros, Ciudad, Performance o Noticias.

Centrando el ejemplo en Noticias, suponiendo que solo se quiere anunciar a través del tablón de noticias de Teatro Accesible. La manera de hacerlo antes de la entrada en vigor del gestor de contenidos seguía los siguientes pasos.

1. Entrar en la web de administración de MySQL que alberga el servidor Apache que se puede ver en la Figura 5.1.

2. Seleccionar la tabla de Noticias (news) e Información de Noticias (newsinfo).
3. Rellenaremos cada uno de los campos de estas tablas “a mano”.
4. Guardamos los datos.

city

content

contentinfo

front

frontinfo

gallery

galleryinfo

help

helpinfo

language

link

linkinfo

news

newsinfo

performance

id	visible	date
1	1	2011-12-27
2	1	2011-12-14
4	1	2011-12-28
5	1	2011-12-30
7	1	2012-01-05

id	newsId	languageId	holder	abstract	text
6	1		1	Ocio compartido con los peques de la casa. Teatro accesible para niños y adultos estas Navida...	Teatro accesib para niños y adultos estas Navida..
7	2		1	Estrenamo temporada accesible El teatro accesible ya no es cosa de una sola sesi...	El teatro accesib ya no es cosa

Figura 4.6: Captura de pantalla de la base de datos.

Una vez que el usuario reclamara la sección de Noticias, la propia página web al renderizar esta parte actualiza (debido a su implementación en código) automáticamente el contenido y aparecería la nueva noticia que hemos guardado en la base de datos.

La dificultad que entraña este proceso se divide en varias partes:

- El número de identificación debe ponerse correctamente siempre.
- Al manejar varias tablas para una misma sección se debe tener claro los números de identificación de cada tabla, con más de 3 tablas puede ser muy complejo y provocar errores en la página web.
- Se corre el peligro de eliminar o modificar datos que no se quieren editar al trabajar directamente sobre la página de administración de la base de datos.
- De cara a las personas que no estén familiarizadas con el entorno, administrar la página puede ser algo complejo.

Con este ejemplo se ha podido ver mejor la necesidad y motivación de este proyecto: la creación de un gestor de contenidos web para el sitio web de Teatro Accesible.

#### 4.4 Catálogo de requisitos

Los requisitos se incluyen y se puede consultar en el ANEXO I del presente documento.

Se capturaron a través de entrevistas con el cliente (Aptent) y con el análisis del sistema inicial. Los requisitos se han clasificado en los siguientes tipos:



- Requisitos de datos
- Requisitos de aplicación
- Requisitos de interfaz
- Requisitos de funcionalidades
- Requisitos de rendimiento
- Requisitos de seguridad

Se presentan a través de dos tablas genéricas donde si se trata de un requisito de dato (ver Tabla 1) indica el número con el que ha sido clasificado, su nombre, el tipo de requisito al que pertenece y los atributos que tiene el elemento. Si se trata de otro tipo de requisito la tabla (ver Tabla 2) indicará el número del requisito, el nombre, su descripción y comentarios sobre el requisito si procede.

Número del requisito	Nombre del Requisito
Tipo	Atributo 1
	Atributo 2

Tabla 1: Tabla genérica de requisitos de datos

Número del requisito	Nombre del Requisito
Descripción	Descripción del requisito
Comentarios	Comentarios sobre el requisito en cuestión

Tabla 2: Tabla genérica de requisitos

### 4.5 Casos de uso

La exploración de los requisitos es vital para construir de forma satisfactoria un sistema que cubra las necesidades de nuestro cliente. Este hecho hace crítico conocer el comportamiento de los usuarios en su interacción con el sistema.

Mediante los casos de uso se describe desde el punto de vista del usuario, el trabajo del sistema para ofrecer una determinada funcionalidad. Por un lado, los modelos de casos de uso se centran en la representación de los requisitos que ha de cumplir el sistema frente a la utilización de los usuarios [31].

Por otro, a través de los diagramas de casos de uso se muestran las interacciones entre usuario y los límites de cada uno de los subsistemas que forman parte del sistema. Esta interacción queda reflejada en el Manual de Usuario que se presenta en el capítulo 7 del presente documento.

En el diagrama de la Figura 4.7 se presenta el caso de uso global que describe todas las interacciones de los usuarios con el software [32]. Podemos observar que para cualquier acción, se deberá estar autenticado y que por lo tanto se ha de cumplir antes con los requisitos de seguridad de la plataforma.

Se recogieron distintos casos de uso con una clara jerarquía. El caso de uso del que dependen todos los demás es el de “Autenticación de usuario”, sin el cual no se puede acceder a los demás y lo que provoca que se cumpla el objetivo del entorno de administración que se describe en el Capítulo 1. A raíz de éste surgen los tres casos de uso principales de los que dependerán la mayoría:

- Gestión de contenido de programación
- Gestión de idiomas
- Gestión de contenido adicional

La manera de representar los casos de usos puede observarse en la Tabla 4 donde podemos ver el número de clasificación del caso de uso, su nombre, su descripción, los requisitos de la plataforma que están relacionados, las condiciones del antes y el después de la acción que se genera en cada caso y la excepción que puede ocurrir en el proceso que genera cualquier caso de uso. Esta forma de representar los casos de uso nos permite una visión general de las acciones y de esta forma se observa mejor la relación entre las distintas partes del proyecto.

Número de clasificación	Nombre del caso de uso	
Descripción		
Requisitos Relacionados		
Precondición		
Postcondición		
Observaciones		
Secuencia de pasos	Paso	Acción
	1	Acción 1
	2	Acción 2
Secuencia de excepción	Paso	Excepción
	1	Excepción generada

Tabla 4: Tabla genérica de casos de uso

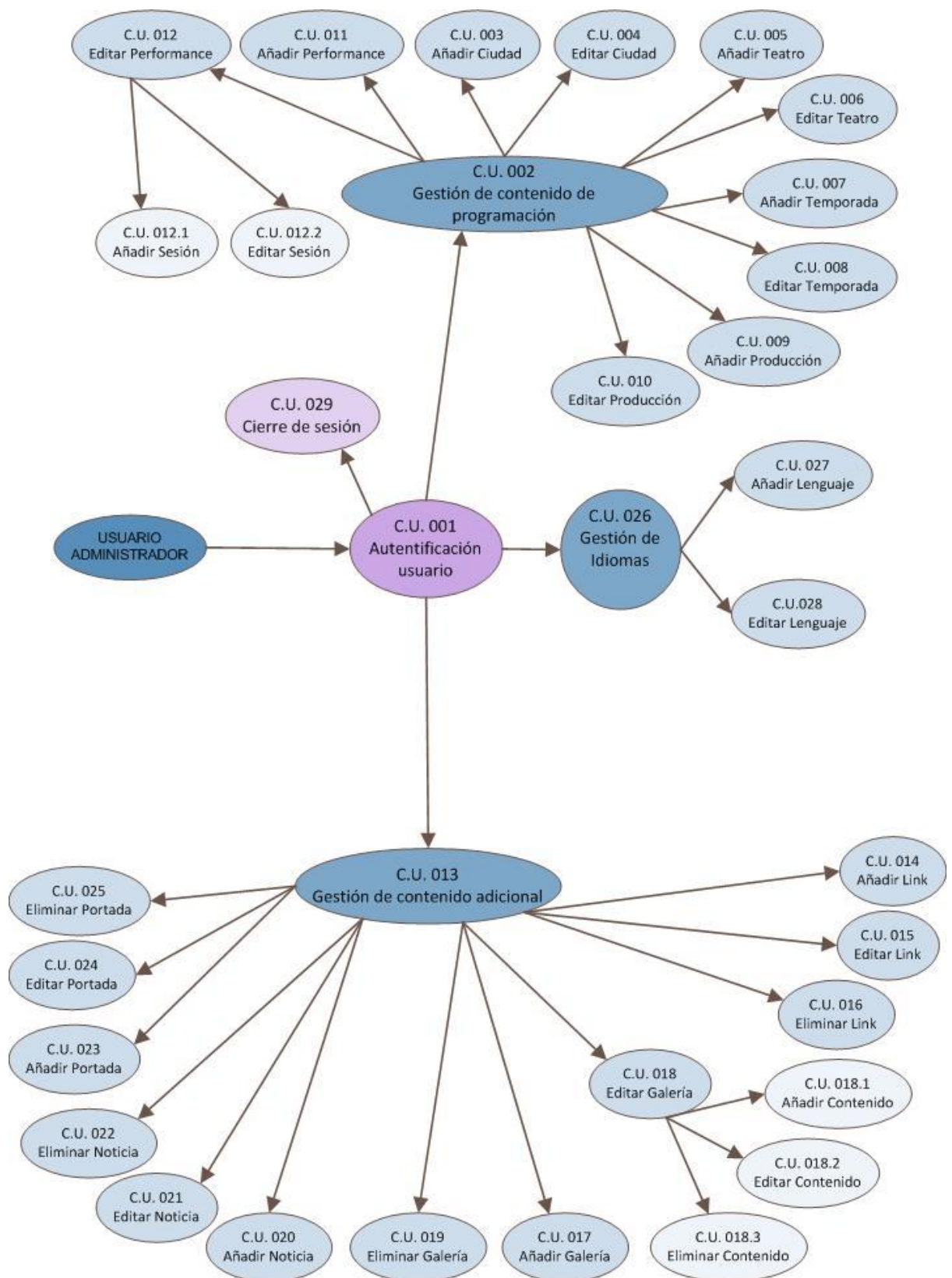


Figura 4.7: Mapa global de Casos de Uso

# Capítulo 5

## Diseño e implementación

### *5.1 Introducción*

Una vez realizado el estudio tecnológico y habiendo seleccionado los lenguajes de programación a utilizar y la arquitectura que se desea mantener para realizar el desarrollo del CMS a través de una aplicación web, se da paso en este Capítulo a la explicación del Diseño e Implementación que se ha elegido para cumplir con los requisitos que se han especificado a lo largo del documento del proyecto.

Tal y como se ha explicado en el Capítulo 4 y como se observa en la Figura 5.1, el proyecto está formado por tres módulos diferenciados, correspondiéndose cada uno de ellos a un componente diferente: Modelos, Vistas y Controladores. Gracias a la utilización del framework Yii estos componentes tienen una clara organización en el entorno de programación. Los paquetes principales que contienen las clases que componen la aplicación del CMS son tres: “Models”, “Views” y “Controllors” que siguen como se ha explicado anteriormente un modelo MVC.

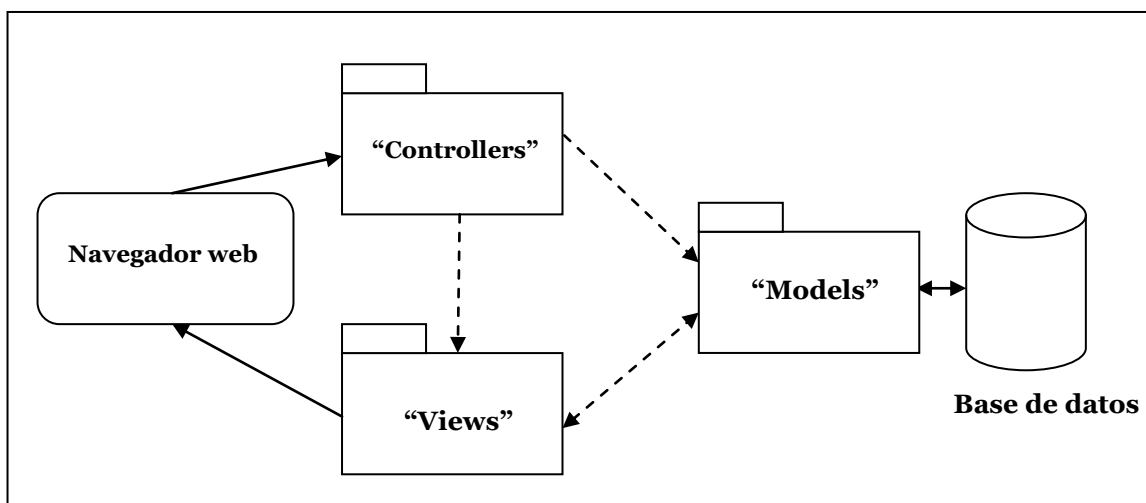


Figura 5.1: Diagrama de paquetes del proyecto para la arquitectura MVC seleccionada

Para profundizar acerca del contenido de estos paquetes, se muestran en los siguientes apartados las clases que los componen con información sobre la función de cada una de ellas. En él se especifican las clases de cada uno de estos paquetes así como los diferentes métodos o funciones que se desarrollan en cada una.

## 5.2 Diseño de clases

El siguiente apartado recoge la documentación del conjunto de clases y componentes que forman la aplicación web del CMS que se desarrolla en este proyecto. Se ha de tener en cuenta tal como se ha explicado en Capítulo de Análisis que al utilizarse el framework Yii se introducen en este modelo MVC variantes que provocan un mejor funcionamiento de este modelo a demás de ayudar al desarrollo del CMS. Estas variantes pueden ser “widgets” o “layouts” que mejoran el rendimiento de una aplicación web como se refleja en la Figura 4.2.

### 5.2.1 Modelos

Los modelos son la representación conceptual de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio)

Los diagramas de clase son el pilar básico del modelado con UML [38], siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). En Figura 5.2 puede verse el modelado UML de las clases modelo de la aplicación web del CMS.

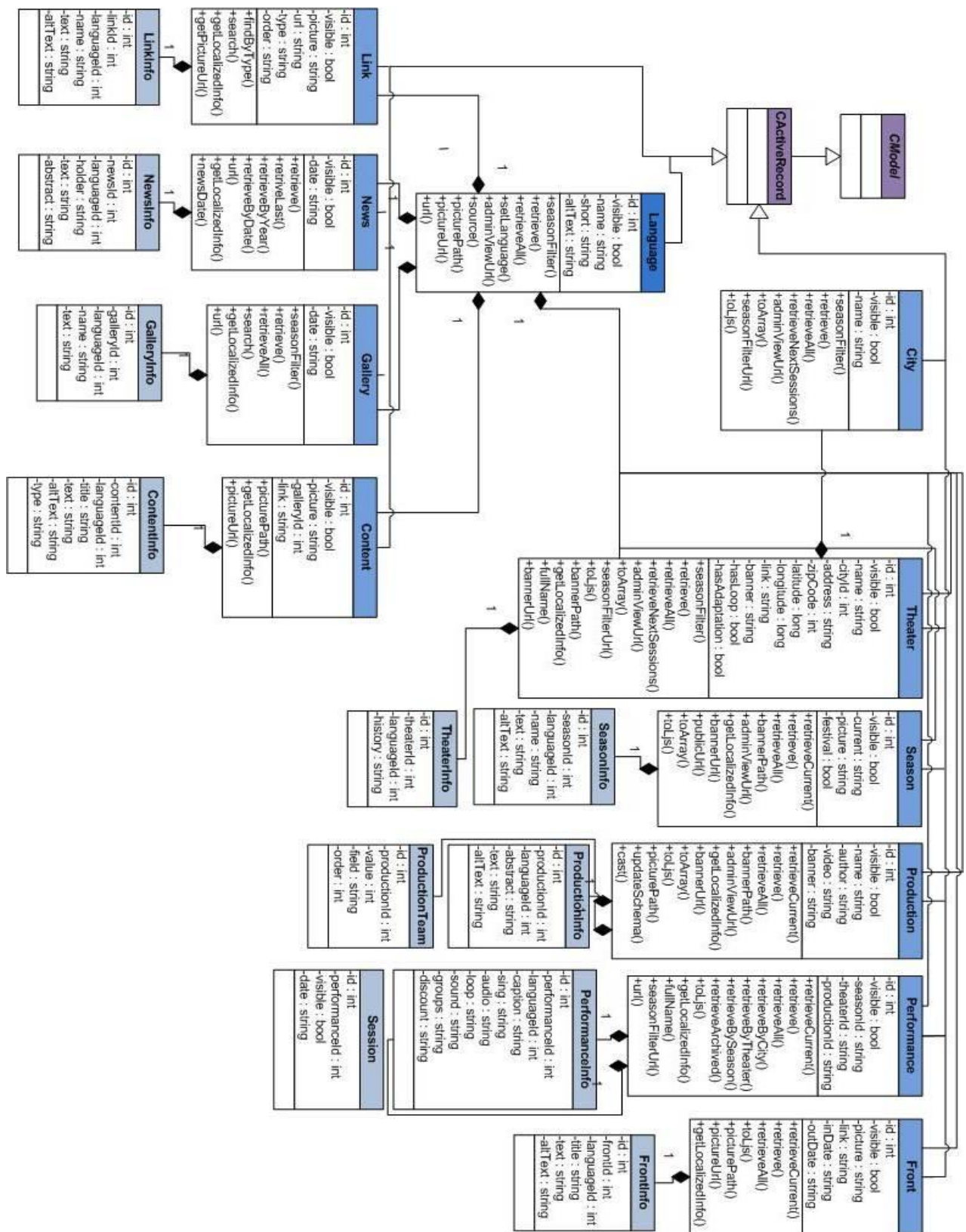


Figura 5.2. Mapa de la arquitectura de clases modelo.

## 5.2.2 Vistas

A lo largo del presente epígrafe se describe cada una de las vistas de la aplicación web. Una vista es un script PHP que contiene elementos de la interfaz de usuario.

Una vista puede contener expresiones PHP, pero es recomendable que estas expresiones no modifiquen los datos del modelo y se mantengan relativamente simples. Es también recomendable que la gran parte de la lógica se encuentre en el modelo y no en la vista para conseguir una mayor separación entre los componentes.

Las vistas se encuentran localizadas dentro de la carpeta `/protected/views`. Como medida de diferenciación, los nombres de los scripts PHP que contienen vistas parciales comienzan con un guión bajo (`_`).

### **`/site/index.php`**

Muestra el mensaje de bienvenida de la aplicación.

### **`/site/login.php`**

Muestra el formulario para iniciar sesión.

Todas las vistas creadas para este proyecto se encuentran dentro de `views` en una carpeta que sigue el patrón de `"admin+(elemento)"`, es decir, en el caso de ciudades sería **`view/admincity/`**, para portadas **`view/adminfront/`** y así sucesivamente. Cada una de estas carpetas contiene una serie de clases que se describen a continuación con los mismos métodos para todas aunque se utiliza una para cada elemento para evitar fallos en los llamamientos a métodos y poder ser más específicos y ordenados a la hora de estructurar la arquitectura del proyecto.

### **`/_form.php`**

Muestra el formulario a rellenar para la creación o modificación de un elemento.

### **`/_grid.php`**

Muestra una tabla con todos los elementos de la tabla que el usuario ha seleccionado.

### **`/_menu.php`**

Muestra el menú superior para llamar a las acciones de Crear, Eliminar o Editar.

### **`/create.php`**

Renderiza la clase `_form.php` con el formulario vacío.

### **`/index.php`**

Ofrece el listado de todos los elementos que forman parte de la tabla con la que se ha decidido trabajar.

### **`/read.php`**

Muestra todos los atributos del elemento guardado o seleccionado.

### **`/update.php`**

Renderiza la clase `_form.php` con el formulario relleno con los datos ya guardados del elemento seleccionado.

### 5.2.3 Controlador

La aplicación crea un controlador cuando recibe una petición de usuario destinada a dicho controlador. Cuando un controlador se ejecuta, realiza la acción solicitada utilizando los modelos necesarios y las vistas apropiadas.

Una acción, en su forma más básica, es un método del controlador cuyo nombre comienza por *action*.

#### **/protected/controllers/AdminController**

Clase abstracta que maneja las peticiones relacionadas con la administración y define las funciones comunes junto con las reglas de acceso.

Las funciones que se desprenden de las clases de administración son:

- **accessRules ( )**  
Reglas de control de acceso.
- **ActionIndex ( )**  
Muestra la página de inicio del portal de administración.
- **ActionLogout ( )**  
Desconecta al usuario y muestra la página de inicio.
- **Filters ( )**  
Filtros para las acciones.
- **actionCreate( )**  
Crea un nuevo elemento.
- **actionRead( )**  
Muestra la información del elemento solicitado.
- **actionUpdate( )**  
Actualiza el modelo solicitado. Si la actualización es satisfactoria se redirige a la página de la vista.
- **actionDelete( )**  
Elimina el elemento solicitado.

#### **/protected/controllers/AdminCityController**

Controlador que maneja las peticiones para administrar el modelo City.

#### **/protected/controllers/AdminTheaterController**

Controlador que maneja las peticiones para administrar el modelo Theater.

#### **/protected/controllers/AdminSeasonController**

Controlador que maneja las peticiones para administrar el modelo Season.



#### **/protected/controllers/AdminProductionController**

Controlador que maneja las peticiones para administrar el modelo Production.

#### **/protected/controllers/AdminPerformanceController**

Controlador que maneja las peticiones para administrar el modelo Performance.

#### **/protected/controllers/AdminNewsController**

Controlador que maneja las peticiones para administrar el modelo News.

#### **/protected/controllers/AdminGalleryController**

Controlador que maneja las peticiones para administrar el modelo Gallery.

#### **/protected/controllers/AdminFrontController**

Controlador que maneja las peticiones para administrar el modelo Front.

#### **/protected/controllers/AdminLinkController**

Controlador que maneja las peticiones para administrar el modelo Link.

#### **/protected/controllers/AdminLanguageController**

Controlador que maneja las peticiones para administrar el modelo Language.

#### **/protected/controllers/SiteController**

Controlador que maneja las peticiones para administrar la página principal. Entre sus métodos más importantes están:

- `actionIndex()`  
Muestra la página de inicio del portal de administración.
- `actionLogin()`  
Muestra la página de login.
- `actionLogout()`  
Desconecta al usuario y muestra la página de login.

### **5.2.4 Layouts**

Un esquema o layout es un tipo de vista especial que se utiliza para decorar las vistas. Puede contener partes de la interfaz de usuario que son comunes a través de muchas vistas.

#### **/layouts/main.php**

Plantilla utilizada en las vistas de la aplicación.

### **5.2.5 Widgets**

Un widget es un componente con propósitos visuales principalmente. Los widgets se embeben en los scripts de vista para generar interfaces de usuario más complejas y contenidas en los propios widgets. De esta forma, se consigue una mayor reusabilidad de los componentes de la interfaz de usuario. Así, mediante un widget es posible definir la cabecera y el pie de

página para todas las páginas. Por otro lado, un widget permite generar una vista basada en los datos del modelo.

Los widgets y sus vistas se localizan, respectivamente, en las carpetas `/protected/widgets` y `/protected/widgets/views`.

**widgets/FooterWidget**

Implementa el pie de página.

**widgets/HeaderWidget**

Implementa la cabecera de las páginas.

**widgets/AdminMenuWidget.php**

Implementa el menú de elementos con los que interactuar (columna izquierda)

**5.3 Implementación de la aplicación web**

Una vez presentados los elementos que componen las vistas de la aplicación, a continuación se recogen ejemplos de las diferentes secciones del CMS, presentando su interfaz, el diseño de su base de datos y mostrando su uso.

La aplicación web del CMS tendrá las características solicitadas por la empresa/cliente pudiendo así realizar la actualización de la base de datos sin apenas esfuerzo y así, conseguir un mantenimiento continuo de los contenidos del sitio web de todos sus aspectos con facilidad.

El CMS constará de cuatro partes principales como vemos en la Figura 5.3:

- Menú de navegación de administración
- Base de datos de la aplicación
- Tabla de atributos
- Interfaz de autenticación

HOME

Logout

Menú

Ciudades

Programación

Ciudades

Teatros

Temporadas

Producciones

Representaciones

Contenidos

Noticias

Galerías

Portadas

Enlaces

Otros

Traducciones

Crear | Editar

ID	Nombre	Visible
<input type="checkbox"/>	1 Madrid	SI
<input type="checkbox"/>	2 Barcelona	SI
<input type="checkbox"/>	3 Valencia	SI
<input type="checkbox"/>	4 Santa Cruz de Tenerife	NO
<input type="checkbox"/>	5 Mérida	SI
<input type="checkbox"/>	6 Rivas-Vaciamadrid	NO
<input type="checkbox"/>	7 El Escorial	SI
<input type="checkbox"/>	8 Almagro	SI

Teatro Accesible

Aptent

W3C CSS

W3C XHTML 1.0

Figura 5.3. Vista principal del CMS.

- Menú de navegación de administración

Es el menú a través del cual se puede elegir qué tabla de la base de datos quiere modificarse. Seleccionando cualquier elemento de la columna de la izquierda de la Figura 5.3 accedemos a la tabla para poder trabajar con ella de manera rápida y sencilla.

- Base de datos de la aplicación

El sitio Web de Teatro Accesible esta creada de manera que existe una conexión entre distintas tablas de datos para que a la hora de mostrar la información en la web sea mucho más sencillo y eficaz hacerlo.

Con motivo de albergar distintas traducciones en el sitio web (castellano, catalán, inglés...), todo lo referente a datos de los que se pueda realizar una traducción se almacena en tablas que heredan de las principales. Estas tablas van acompañadas de la palabra “info” tras su nombre principal. Se encuentran albergadas en el servidor de MySQL de la aplicación.

Si por ejemplo hay una noticia nueva, la tabla “news” solo alberga atributos en los que la traducción no tiene sentido, como por ejemplo: Id, visible o fecha.

Sin embargo en la tabla “newsinfo” podemos ver atributos como: holder, text, altText que se refieren al título y texto de la noticia, el cual sí que podría tener traducción.

Es por tanto, es importante entender que cuando se selecciona una de las variables del menú de la izquierda que muestra la Figura 5.4 no solo se accede a la tabla que el nombre indica sino que se puede estar trabajando con más de una tabla.

Entre las tablas a las que se permite el acceso como administrador están:

Menú

Programación

Ciudades

Teatros

Temporadas

Producciones

Representaciones

Contenidos

Noticias

Galerías

Portadas

Enlaces

Otros

Traducciones

- Ciudades (city)
- Teatros( teather y teatherinfo)
- Temporadas (season y seasoninfo)
- Producciones (production y productioninfo)
- Representaciones(performance y performanceinfo)
- Noticias (news y news info)
- Galerías (gallery y galleryinfo)
- Portadas (front y front info)
- Enlaces (links y linksinfo)
- Traducciones (language)

Figura 5.4. Menú de elementos.

- Tablas de atributos

Al pulsar cualquiera de las opciones anteriores se despliega en la columna derecha de la interfaz (o centro de la pantalla, Figura 5.5) una tabla que refleja los atributos más importantes (no muestra todos) de esa tabla en concreto, para poder así identificar el elemento que se busca para editar o eliminar por ejemplo.

Cada una de estas tablas tiene un pequeño menú en su parte superior donde se puede elegir entre CREAR un nuevo elemento, EDITAR uno existente (seleccionando previamente el elemento) o ELIMINAR el elemento que se halla seleccionado.

Además existe otra opción llamada DETALLES a la que se puede acceder al pulsar el ID del elemento del cual se quieren ver todos los atributos. Incluso atributos que en la tabla principal no aparecen.

Crear   Editar   Eliminar				
	Id	Visible	Nombre	Abreviatura
<input type="checkbox"/>	1	SI	Español	es
<input type="checkbox"/>	2	NO	English	en

Figura 5.5. Tabla de atributos del elemento seleccionado.

Para tener un correcto diseño e implementación de la base de datos, se definieron las restricciones y reglas de integridad referencial. En función de restricciones, no todas las tablas pudiesen tener las mismas opciones, es decir que sólo en algunas se pueda decidir eliminar para asegurar así que el sitio web de Teatro Accesible no albergue fallos al manejar el CMS ya que al borrar un elemento de una tabla se puede estar modificando otras tablas y producir errores muy difíciles y tediosos de solucionar.

Si por ejemplo se borra una de las ciudades de la base de datos hace que todas las tablas que necesiten el atributo de Identificador de Ciudad no lo tengan y se cree una incongruencia que desemboca en un error, muy difícil de solucionar, en la web.

Entre las tablas de las que pueden borrarse elementos se encuentran:

- Representaciones
- Noticias
- Galerías
- Portadas
- Enlaces
- Traducciones

Dentro del menú superior de la tabla cuando se pulsa una de las tres opciones ocurre lo siguiente:

CREAR: se actualiza la página mostrando un formulario (Figura 5.6) en el que se podrán rellenar los atributos de las tablas que componen elemento a crear. Se exigirán ciertos campos y formatos para evitar posibles errores al leer la base de datos. Al pulsar el botón de “Crear” se habrá creado un nuevo elemento en la tabla que la página web interpretará y mostrará o no dependiendo de si se ha seleccionado la opción de “visible” o no.

Los campos marcados con \* son necesarios.

Visible ☐

Nombre \*

Abreviatura \*

Texto alternativo

Imagen  No se ha seleccionado ningún archivo

Crear

Figura 5.6. Formulario para la creación de un nuevo elemento.

MODIFICAR: muestra el mismo formulario (Figura 5.7) que CREAR pero cargando previamente el contenido que ya de por si tiene el elemento para que así se pueda ver qué atributo del elemento se desea modificar. El botón de “Actualizar” modifica los datos que han sido cambiados en la base de datos y mantiene los que no se editaron.

Los campos marcados con \* son necesarios.

Visible ☒

Nombre \*

Abreviatura \*

Texto alternativo

Imagen

Nueva imagen  No se ha seleccionado ningún archivo

Actualizar

Figura 5.7. Formulario para editar un elemento.

ELIMINAR: tras seleccionar un elemento de la tabla se podrá eliminar al pulsar en el menú superior eliminar, no sin antes una comprobación de seguridad para evitar posibles errores por parte del administrador que se muestran en la Figura 5.8.

	Id	Visible	Nombre
<input checked="" type="checkbox"/>	1	SI	Español
<input type="checkbox"/>	2	NO	English
<input type="checkbox"/>			

Mensaje de la página localhost:

¿Eliminar el elemento seleccionado?

Aceptar Cancelar

Figura 5.8. Evento de confirmación para eliminar.

**DETALLES:** a esta opción se puede acceder al pulsar el ID del elemento del cual se quieren ver todos los atributos. Muestra todos los atributos del elemento (Figura 5.9).

<input checked="" type="checkbox"/>	5	SI	Teatro María Guerrero	Tama
Detalles: Teatro María Guerrero				
Nombre	Teatro María Guerrero			
Visible	1			
Dirección	Tamayo y Baus. 4			
Codigo Postal	28004			
Latitud	40.422779			
Longitud	-3.693056			
Link	http://cdn.mcu.es/programacion/temporada-actual/maria-guerrero/			
Banner	No asignado			
Bucle Magnetico	1			
Adap. movilidad	1			

Figura 5.9. Pantalla de detalles del elemento seleccionado

- Interfaz de autenticación**

El perfil de administración fue exigido por el cliente con motivo de una mayor seguridad a la hora de administrar el sitio web. El hecho de que la página web de administración de contenidos sea parte o este en el mismo servidor del sitio web de Teatro Accesible obliga de alguna manera a pasar por un proceso de autenticación (login) para aquel que quiera entrar en la base de datos como se muestra en la Figura 5.10.

Este login es el que usualmente se utiliza en páginas web, para poder acceder a cualquiera de las tablas de la base de datos antes debe iniciarse sesión con un USER y un PASSWORD. Si este no coincide con las credenciales almacenadas por defecto se deniega el acceso.

En caso contrario se le da acceso al menú desde el que podrá trabajar sin problema alguno. Se dispone también de un Logout para salir de la sesión de administrador.

HOME

Logout

Teatro Accesible

Usuario \*

Password \*

Recordar ☐

Login

Teatro Accesible

Aptent

W3C

CSS

W3C

XHTML

Figura 5.10. Página de autenticación.

# Capítulo 6

## Pruebas

### **6.1    *Introducción***

La finalidad del presente capítulo es hacer una validación de la aplicación desarrollada y confirmar que se da un correcto funcionamiento de la aplicación cumpliendo los requisitos. Para garantizar este correcto funcionamiento, se han definido y llevado a cabo diferentes tipos de pruebas para detectar todos aquellos errores que durante la codificación y programación de las diferentes etapas no han podido detectarse. Este punto es fundamental en el desarrollo de una aplicación de estas características, ya que hemos de conseguir una correcta visualización y una correcta respuesta a los posibles errores ante el usuario que la use.

Para detectar estos errores, se han llevado a cabo tanto pruebas unitarias como de integración que a continuación se describen.

### **6.2    *Pruebas unitarias***

Las pruebas unitarias son aquellas que permiten analizar los diferentes módulos del sistema de forma individual. Como módulo podríamos entender cada una de las tablas de la base de datos que maneja el CMS. En el desarrollo de esta aplicación, cada vez que se acababa un módulo, éste se ha sometido a una serie de pruebas para encontrar los posibles errores. Así como pruebas siguiendo las indicaciones de la aplicación para ver que realiza correctamente su función.

Las diferentes pruebas que se han realizado dentro de esta categoría son las siguientes:

- Validación de campos vacíos: Se ha comprobado el correcto funcionamiento de la definición de las tablas y restricciones de la base de datos subyacente de la aplicación tal que al realizar alguna acción determinada, deben estar todos los campos obligatorios rellenos. Y que un mensaje de error indicando qué campos son los que faltan por rellenar.

- Validación del formato de los datos de entrada: Se ha comprobado la correcta definición en el tipo de datos y comprobaciones de la base de datos subyacente de la aplicación tal que aparte de estar todos los campos obligados rellenos, tal y como indica el punto anterior, también que estos campos tengan el formato correcto pedido para que no haya problemas a la hora de insertar los datos en la base de datos. Y que un mensaje de error indicando qué datos son los que tienen un formato incorrecto.

- Validación de la información en la base de datos: Se ha comprobado la correcta definición de las tablas y restricciones de la base de datos subyacente de la aplicación al insertar, eliminar o modificar cualquier dato en la base de datos, se haga correctamente.

- Correcta visualización por parte del módulo: Se ha comprobado que el módulo en cuestión se visualice bien tanto en Mozilla Firefox como en Google Chrome.

### **6.3 Pruebas de integración**

Estas pruebas son aquellas que se realizan de forma conjunta una vez se tienen todos los módulos desarrollados, es decir, la aplicación ya finalizada.

Lo que se intenta encontrar son aquellos errores derivados de acciones como el paso de parámetros entre diferentes páginas web de la aplicación o el envío y recuperación de los datos con formularios.

A continuación, se detallan las pruebas realizadas:

- Inicio y cierre de sesión: se ha comprobado el correcto funcionamiento de las credenciales a la hora de acceder o salir de la aplicación como administrador

- Creación: Se ha comprobado la correcta definición las restricciones de integridad referencial de la base de datos subyacente de la aplicación que una vez el usuario ha iniciado sesión y todos los campos obligatorios son rellenos correctamente se efectúa correctamente insertando los datos introducidos en la base de datos.

- Eliminación: Se ha comprobado tal la correcta definición las restricciones de integridad referencial de la base de datos subyacente de la aplicación tal que una vez el usuario ha iniciado sesión y se ha seleccionado el elemento que se quiere eliminar se efectúa correctamente el borrado de los datos de la base de datos correspondientes al elemento en cuestión.

- Modificación: Se ha comprobado tal la correcta definición las restricciones de integridad referencial de la base de datos subyacente de la aplicación tal que una vez el usuario ha iniciado sesión y se ha seleccionado el elemento que se quiere editar se efectúa correctamente la modificación de los datos de la base de datos correspondientes al elemento en cuestión.

### **6.4 Catálogo de pruebas**

Todo este conjunto de pruebas de funcionamiento ha sido creado con el propósito de revisar y garantizar el buen funcionamiento del CMS desarrollado e implementado. Tras el desarrollo del mismo se ha revisado que éste ha cumplido cada una de las pruebas definidas en el ANEXO III.



En la Tabla 2 podemos apreciar en detalle las pruebas ya que cada prueba tiene un claro objetivo donde se valora la criticidad, el proceso de evaluación y los objetivos secundarios que deben cumplirse para que la prueba se dé por superada. Además se analiza que ocurre si la prueba se realiza correctamente o incorrectamente y relaciona cada prueba con el Caso de Uso correspondiente.

Prueba		Numero de prueba		Caso de Uso		Número Caso de uso	
Objetivo							
Criticidad		Alta		Media		Baja	
Punto de medición							
Contenido a chequear							
Ejecución correcta				Ejecución incorrecta			
Juego de Datos							
Validación		Campo		Ejecución			
Realizada	Mensaje			Correcta		Incorrecta	

Tabla 2: Tabla genérica de evaluación de pruebas

En cuanto a las pruebas que tienen que ver con el almacenamiento de datos en la base de datos a través del CMS se realiza lo que se llama un “Juego de Datos” [42] donde cada “Campo”(o atributo de una lista de la base de datos) atraviesa dos fases, una de validación y otra de ejecución.

En la fase de validación se muestran las reglas que hay que cumplir para cada Campo al rellenar las tablas de la base de datos para que no existan errores. Se muestra qué tipo de regla existe en el apartado de “Realizada” y el “Mensaje” asociado.

A su vez, la fase de Ejecución muestra con ejemplos la manera correcta e incorrecta de rellenar los atributos de la tabla seleccionada.

Las pruebas que se desglosan en el ANEXO III son fruto de una exhaustiva búsqueda de errores que se ha ido llevando a cabo durante todo el desarrollo del CMS. De esta forma se ha asegurado que el funcionamiento de la aplicación web del CMS se mantiene estable ante cualquier imprevisto o mal uso de aquel que lo maneja.

# Capítulo 7

## Manual de usuario

### 7.1 *Introducción*

Con el objetivo de cumplir con los requisitos de este proyecto se presenta a continuación el manual de usuario. Este manual le permitirá aprender a utilizar todas las funcionalidades básicas del gestor de contenidos web de la plataforma “Teatro Accesible”.

Esta aplicación es una herramienta para la gestión de la base de datos del sitio web [www.teatroaccesible.com](http://www.teatroaccesible.com) que brinda la posibilidad de crear, modificar y eliminar elementos de la base de datos para que la página web de la plataforma actualice su contenido con facilidad y rapidez.

Será necesario tener las credenciales de administrador para acceder a la aplicación y poder así utilizarla.

### 7.2 *Instalación*

Al tratarse de una aplicación web, los usuarios pueden acceder a ella mediante un navegador. Por lo tanto sólo es necesario que el usuario tenga instalado en su equipo un navegador web como Mozilla Firefox, Google Chrome.

### 7.3 *Acceso*

- Iniciar sesión

Acceda a la URL de la aplicación: *Dirección\_IP\_Servidor:8081/TeatroAccesible/site/login* (por seguridad y a petición de la empresa cliente no se revela en la memoria la dirección IP

para acceder al servidor que contiene la aplicación web del CMS) y una vez se le muestra el formulario de entrada de la Figura 7.1, introduzca su usuario y contraseña y pulse “Login”.

The image shows a login form for 'Teatro Accesible'. At the top, the text 'Teatro Accesible' is displayed, with 'Teatro' in a large blue font and 'Accesible' in a smaller blue font inside a dark blue rectangular box. Below this, there are two input fields: 'Usuario \*' and 'Password \*', both with red asterisks indicating they are required. To the right of the 'Password \*' field is a checkbox labeled 'Recordar'. At the bottom of the form is a button labeled 'Login'.

Figura 7.1: Formulario de entrada a la aplicación.

7.4 Utilización

En la parte izquierda de la aplicación podrá encontrar un menú en el que podrá seleccionar el tipo de elemento que quiere crear, modificar o eliminar como muestra la Figura 7.2.

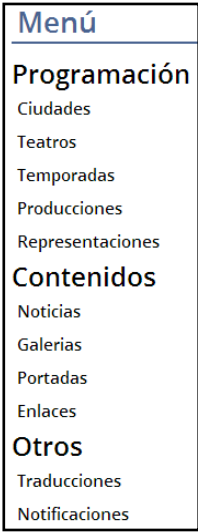
The image shows a vertical menu titled 'Menú'. It is divided into three sections: 'Programación', 'Contenidos', and 'Otros'. Under 'Programación' are the items: Ciudades, Teatros, Temporadas, Producciones, and Representaciones. Under 'Contenidos' are: Noticias, Galerías, Portadas, and Enlaces. Under 'Otros' are: Traducciones and Notificaciones.

Figura 7.2: Listado de elementos de la base de datos.

Una vez seleccione el elemento deseado aparecerán a su derecha una serie de atributos en una lista (Figura 7.3). Éstos son algunos de los atributos existentes en la lista de este elemento de la base de datos.

ID	Nombre	Visible
<input type="checkbox"/>	1 Madrid	SI
<input type="checkbox"/>	2 Barcelona	SI
<input type="checkbox"/>	3 Valencia	SI
<input type="checkbox"/>	4 Santa Cruz de Tenerife	NO
<input type="checkbox"/>	5 Mérida	SI
<input type="checkbox"/>	6 Rivas-Vaciamadrid	NO
<input type="checkbox"/>	7 El Escorial	SI
<input type="checkbox"/>	8 Almagro	SI

Figura 7.3: Menú de selección de elemento

Una vez hecho este proceso podrá elegir entre las siguientes tres opciones en el menú superior de la pantalla (Figura 7.4):



Figura 7.4: menú de opciones.

- Crear elementos

Pulse en la opción de “Crear” del menú superior y rellene el formulario que se muestra en la Figura 7.5 y pulse “Guardar”. Recuerde que debe rellenar todos los campos obligatorios (marcados con \*).

Los campos marcados con \* son necesarios.

Nombre \*

Descripción \*

Visible ☐

Actual ☐

Festival ☐

Seleccionar Banner (685x140)  No se ha seleccionado ningún archivo

Figura 7.5: Formulario de creación de un elemento.

- Modificar elemento

Seleccione el elemento que quiere modificar como se muestra en la Figura 7.6. Pulse en la opción de “Editar” del menú superior y rellene el formulario que se muestra en la Figura 7.7, después, pulse “Guardar”. Recuerde que debe rellenar todos los campos obligatorios (marcados con \*)

Id		
<input checked="" type="checkbox"/>	1	Ta
<input type="checkbox"/>	2	Fe

Figura 7.6: Sistema de selección de elementos.

Los campos marcados con \* son necesarios.

Nombre \*

Temporada 2011/12

Descripción \*

Primera temporada de Teatro Accesible en los teatros

Visible

☒

Actual

☐

Festival

☐

Banner \*

season\_1.jpg

Seleccionar Banner  
(685x140)

Seleccionar archivo

No se ha seleccionado ningún archivo

Actualizar

Figura 7.7: Formulario de edición de un elemento.

- Eliminar elemento

Seleccione el elemento que quiere modificar como se muestra en la Figura 7.6. Pulse en la opción de “Eliminar” del menú superior. Le aparecerá una ventana emergente que le pedirá confirmación para eliminar el elemento. Tras pulsar “Aceptar” habrá eliminado el elemento. Esta alerta se muestra en la Figura 7.8.

	Id	Visible	Nombre
<input checked="" type="checkbox"/>	1	SI	Español
<input type="checkbox"/>	2	NO	English
<input type="checkbox"/>			

Mensaje de la página localhost:

¿Eliminar el elemento seleccionado?

Aceptar

Cancelar

Figura 7.8: Ventana emergente de alerta para la eliminación de elementos.

También puede elegir ver los detalles de un elemento en concreto, para hacer esto solo deberá pulsar el “ID” del elemento en el que esté interesado (Figura 7.9) y le aparecerá automáticamente la pantalla de detalles del elemento que seleccionó.

ID		
<input type="checkbox"/>	1	M
<input type="checkbox"/>	2	Ba

Figura 7.9: Enlace a los detalles de un elemento.

## 7.5 Cierre de sesión

Una vez dentro de la aplicación, en la parte superior derecha encontrará el texto “Logout” como muestra la Figura 7.10. Púlselo para cerrar su sesión.



Figura 7.10: Texto de cierre de sesión.

# Capítulo 8

## Gestión del proyecto

### **8.1** *Introducción*

El capítulo presenta la estimación de recursos económicos y humanos necesarios y a continuación muestra el plan de utilización de los mismos a lo largo de las diferentes etapas.

La planificación surge de la necesidad de estimar los costes económicos y temporales que requiere el desarrollo de un proyecto así como del control de los factores que pueden alterar su evolución [40]. En base al conjunto de funcionalidades propuestas, se hace imprescindible definir un calendario de trabajo con el objetivo de ordenar la ejecución de cada una de las tareas necesarias para su implementación.

El proceso software es la descripción de las etapas que se siguen durante la ejecución de un proyecto. Su correcta definición permite asegurar una adecuada asignación de recursos y conocer el estado del proyecto en cada momento. Establece el marco de trabajo, tanto técnico como de gestión, en la aplicación de métodos, las herramientas y las personas a las tareas de desarrollo de software [41].

Una metodología de trabajo definida permite que cada nuevo proyecto sea construido en base a la propia experiencia y a la de predecesores. Su utilización permite identificar las causas de los problemas para corregirlos. En la Figura 8.1 se muestra el proceso donde se unen las actividades de desarrollo y gestión. Estas últimas se orientan a controlar el desarrollo del proceso software y a corregir las desviaciones con respecto a los parámetros de calidad establecidos. Los procesos de gestión se ejecutan de manera paralela a las operaciones de desarrollo.

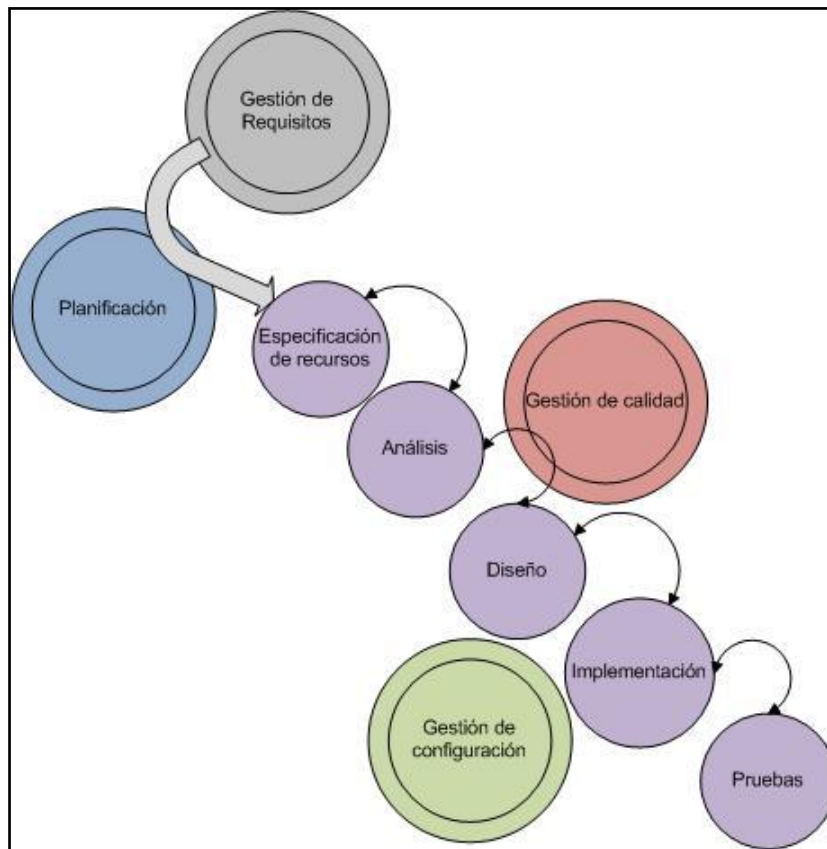


Figura 8.1: Proceso de Desarrollo Software

## 8.2 Metodología

El ciclo de vida es el periodo que comienza cuando se concibe un producto software y termina cuando dicho producto deja de estar disponible. Se divide normalmente en fases que estructuran y organizan las etapas de concepción, desarrollo y mantenimiento del sistema software.

Debido a la naturaleza del proyecto se ha utilizado el ciclo de vida en espiral. En este modelo el proyecto ataca en una serie de ciclos de vida cortos, cada uno de los cuales finaliza con una revisión de software ejecutable.

En la Figura 8.2 se muestra un ejemplo de desarrollo completado en varias iteraciones del ciclo. En cada una de éstas se desarrollan todas las fases del ciclo y se incorporan las experiencias de las entregas anteriores, con sucesivas etapas de especificación, construcción, explotación y revisión para obtener un refinamiento del producto de cara al siguiente ciclo.

Este ciclo de vida es especialmente recomendable cuando se planea hacer múltiples entregas del software. Se emplea en proyectos de gran tamaño o larga duración, en aquellos que necesitan tecnologías muy avanzadas o cuando se requiere experiencia del usuario para refinar el diseño. Las principales motivaciones que han llevado a su elección son las siguientes:

- Complejidad del sistema: debido a la complejidad del proyecto y a los recursos disponibles se ha abordado el desarrollo de cada uno de los componentes de la plataforma por separado en varios ciclos. Esto ha permitido corregir anomalías y desviaciones.



- Riesgo tecnológico: el ciclo de vida adoptado permite asumir ciertos riesgos derivados del desconocimiento tecnológico.
- Dinamismo y adaptabilidad: el ciclo de vida acerca el sistema a la solución de forma progresiva y permite la especialización del equipo de desarrollo en función del componente a implementar.

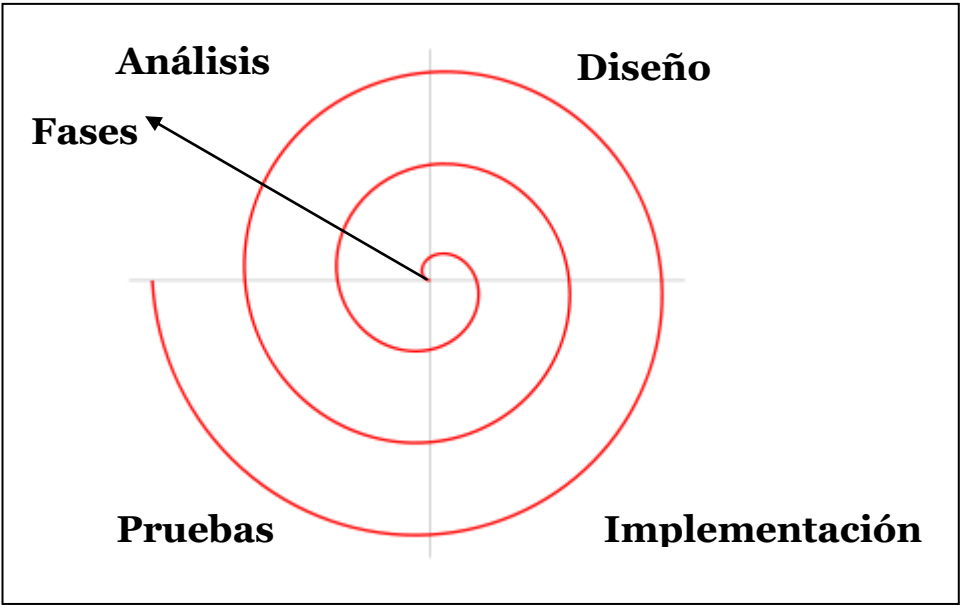


Figura 8.2: Ciclo de Vida en Espiral

8.3 Estimación de recursos

La fecha de inicio del proyecto se establece el día 15 de Abril de 2013 y la fecha de finalización en base a la actual planificación se emplaza el día 7 de Junio de 2013. Entre estas dos fechas se sitúa el desarrollo del proyecto con un total de 446 horas. El reparto de las horas para cada una de las tareas del proyecto se presenta en la Tabla 5.

Tarea		Horas
1	Planificación	80
2	Análisis	88
3	Diseño	80
4	Codificación	152
5	Pruebas	56
6	Despliegue	8
7	Seguimiento	4
Total		446

Tabla 5. Recursos Temporales por Fases del Proyecto

El diagrama de Gantt de la Figura 8.3 muestra la sucesión de cada una de las tareas que componen el desarrollo del proyecto.

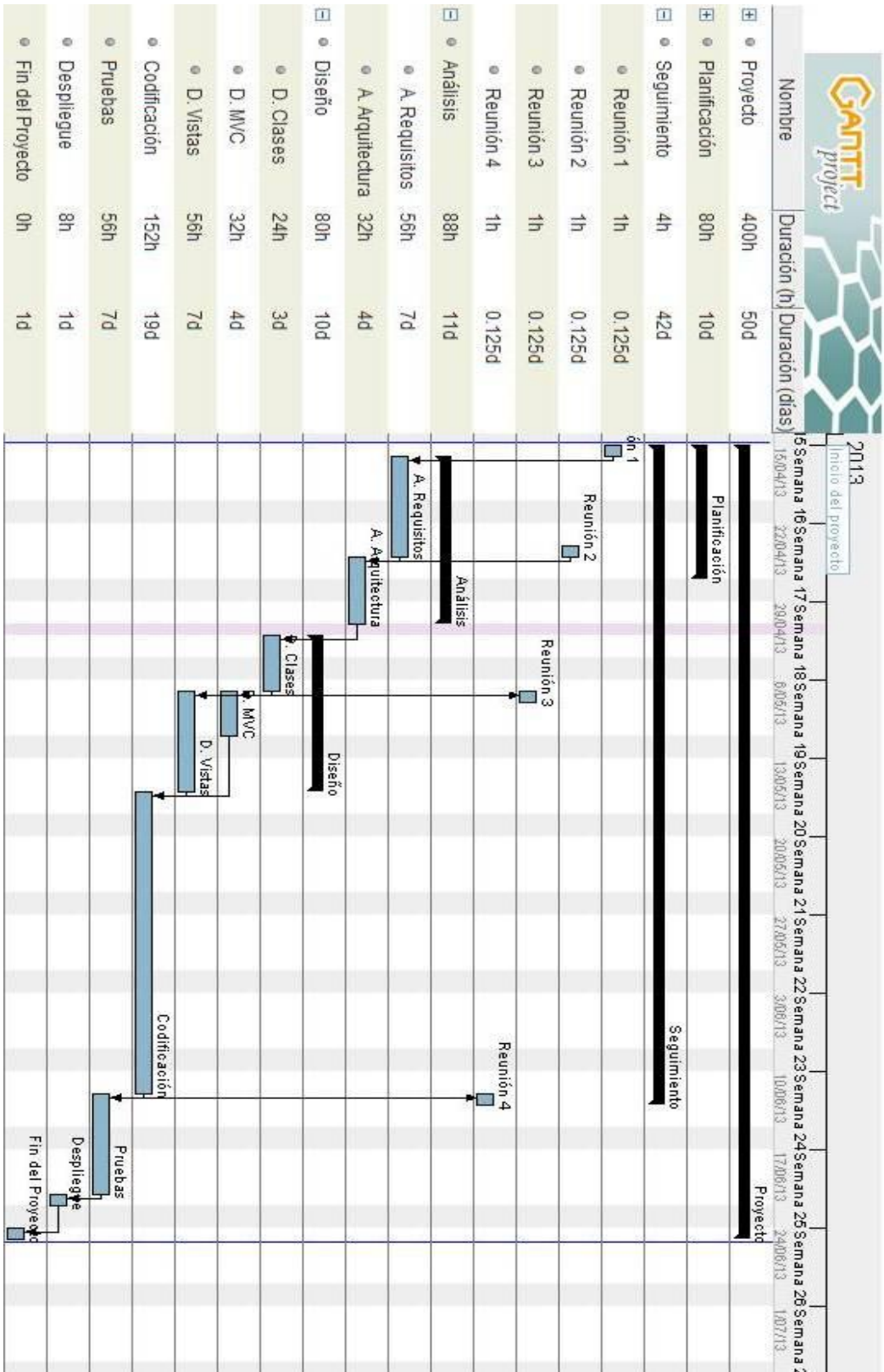


Figura 8.3: Diagrama de Gantt

8.4 Estimación de recursos económicos

En las siguientes secciones se determina el coste asociado a la ejecución del proyecto.

8.4.1 Recursos Materiales

La estimación del coste y de los periodos de amortización de los recursos materiales se han obtenido en base a lo descrito en [RD1777, 2004]. En este documento se establece un plazo de amortización de 8 años para los elementos de tratamiento de información y de 6 años para los programas informáticos. En base a esta información, el coste asociado a cada elemento viene determinado por la siguiente expresión.

Coste = Precio · Coef = Precio · (Unidades · d/a)

Figura 8.4. Expresión para el Cálculo de las Amortizaciones

En la expresión anterior, a y d corresponden al periodo de amortización del elemento y a la duración del proyecto en años, respectivamente. La tabla 6 muestra el desglose de los costes asociados a los recursos materiales.

Concepto	Precio	Coeficiente	Coste
PC Asus Intel Core i5	482,79 €	9,4 %	43,38 €
Conexión F.O. a 100MB	30 €	9,4 %	2,82 €
PC Acer Aspire Intel Core i3	489 €	12,5 %	45,96 €
Microsoft Office 2007	129,99 €	12,5 %	16,24 €
Microsoft Visio 2007	739,00 €	12,5 %	92,37 €
TOTAL			200,77 €

Tabla 6. Recursos Materiales para el Desarrollo del Proyecto.

8.4.2 Recursos Humanos

A partir de los recursos temporales presentados en la sección 8.2 y de los perfiles profesionales descritos a continuación, en la siguiente sección se determina el coste asociado a los recursos humanos. En la Tabla 7 se muestran los perfiles de los profesionales requeridos para la ejecución de las diferentes tareas y los costes de cada uno de ellos.

Concepto	€/Hora	Horas	Coste
Director de Proyecto	50,00 €	158	7900,00 €
Analista	30,00 €	80	2400,00 €
Programador	30,00 €	208	6240,00 €
TOTAL			16.540,00 €

Tabla 7. Recursos Humanos.

Los perfiles profesionales requeridos para desarrollar cada una de las tareas son los siguientes:

- Director de Proyecto: Profesional titulado en Ingeniería Informática /Telecomunicaciones con experiencia. Requiere habilidades como liderazgo, capacidad de comunicación e intermediación en conflictos. Los honorarios de este profesional se sitúan en 50 €/hora.
- Analista: Profesional titulado en Ingeniería Informática /Telecomunicaciones con amplia formación en gestión de proyectos y recursos humanos. Los honorarios de este profesional se encuentran en 30 €/hora.
- Desarrollador de Aplicaciones: Profesional titulado en Ingeniería Informática con conocimientos en PHP, HTML, Java Script, CSS y otras tecnologías relacionadas con la elaboración de contenidos Web. Sus honorarios se sitúan en 30 €/hora.

### ***8.4.3 Costes Totales***

El coste total asociado al desarrollo del proyecto asciende a 16.540,77 € (dieciséis mil quinientos cuarenta euros y setenta y siete céntimos).

# Capítulo 9

## Conclusiones y líneas futuras

### 9.1 *Introducción*

Este último capítulo presenta los resultados obtenidos y conclusiones. Así mismo, se ofrece las líneas futuras de trabajo.

### 9.2 *Conclusiones*

Las conclusiones presentadas a continuación resultan de la confrontación entre los objetivos planteados al inicio del proyecto y los conseguidos tras la finalización del mismo.

Una de las tareas más difíciles a la hora de desarrollar un sistema software es establecer las condiciones de satisfacción para que el trabajo se dé por concluido. Para establecer el grado de éxito del proyecto se recuerdan a continuación los objetivos generales de la plataforma que fueron citados en el Capítulo 1 y los alcanzados finalmente.

Los objetivos iniciales eran:

- Proporcionar un sistema de gestión de contenidos (Content Management System, abreviado CMS) a través de una aplicación web, para que el usuario pueda actualizar los contenidos que desee.
- Proporcionar una política de administración en el gestor de contenidos mediante la creación de un perfil de administrador con privilegios para poder llevar a cabo tareas de administración.
- Dotar al gestor de contenidos de un manual de usuario para su posterior utilización de manera que cualquier usuario pueda usarlo sin dificultad.

Como se ha documentando a lo largo de todos los capítulos, estos objetivos han sido alcanzados con éxito. A raíz de estos objetivos han surgido otros supeditados que se resumen a continuación:

- Se ha desarrollado una plataforma capaz de manejar bases de datos en MySQL sin conocimientos previos sobre este tipo de base de datos.
- Se ha desarrollado un gestor capaz de crear, eliminar o editar elementos de la base de datos además de poder acceder a ellos en cualquier momento.
- Se ha logrado desarrollar una aplicación que salvaguarda la información a través de identificación previo que obliga al usuario a identificarse.

Un punto clave en el desarrollo del proyecto ha sido el trabajo en equipo. La mayor parte de los objetivos han requerido un constante trabajo en equipo además de una comunicación bidireccional entre el cliente y desarrollador.

A la hora de programar o desarrollar una aplicación web de estas características que conlleva una tecnología original es necesario aprender sobre nuevas técnicas, sobre nuevas tecnologías y poseer de antemano un conocimiento sobre el desarrollo web. Durante el desarrollo de este gestor de contenidos he tenido que aprender a programar en lenguajes que no había utilizado antes lo que me ha llevado a alcanzar cierta destreza como programador que de otro modo no hubiera alcanzado. Además el hecho de trabajar con una empresa y de el modo que se ha descrito ha supuesto un gran escalón en el ámbito personal y profesional de cara al futuro.

### **9.3 *Líneas futuras***

En esta sección se enumerarán algunas de las mejoras propuestas para el sistema de gestión de contenidos a través de la aplicación web desarrollado:

- Movilidad de la aplicación a los dispositivos móviles, esto consistiría en que al actualizar la base de datos los usuarios puedan recibir alertas de información sobre lo que se acaba de actualizar en sus móviles mediante ventanas emergentes.
- Que los usuarios puedan suscribirse en concreto a un teatro o ciudad determinada.
- Mejorar los mecanismos de seguridad en la autenticación a través de un captcha, u otras soluciones.
- Definición de una política de administración con distintos tipos de administradores y privilegios. Creación de roles de usuario de administración quedando constancia de lo que cada usuario hace y repartiendo diferentes responsabilidades. Es decir, si hay alguien encargado de las noticias, que solo pueda acceder a ese tipo de elementos en la base de datos.

# Referencias

- [1] Roldan Martínez, D., Valderas Aranda, P., Pastor López, O. (2010), Aplicaciones web. Un enfoque practico. España: Alfaomega, Ra-Ma.
- [2] Lemus J.M., (2007), Desarrollo multinivel para aplicaciones basadas en el web, <http://www.maestrosdelweb.com/editorial/desarrollo-multinivel-para-aplicaciones-basadas-en-el-web/>
- [3] Harkey D., Orfali R., Edwards J., (2000), Cliente/Servidor: Guía de supervivencia (2ª ed.), McGraw-Hill / Interamericana de México
- [4] Centro Nacional de Tecnologías de la Accesibilidad (CENTAC) , (2005), Accesibilidad total audiovisual para personas sordas, ciegas y mayores <http://www.centac.es/es/content/ayna-una-aplicaci%C3%B3n-revolucionaria-para-ciegos-y-sordos>
- [5] Proyecto cofinanciado por: Ministerio de Industria, Turismo y Comercio dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011 y el Fondo Europeo de Desarrollo Regional (FEDER). Proyecto de la Web 4.0 <http://46.4.52.82/cms/index.php>
- [6] Cobo, A., (2005) PHP y MySQL: “Tecnología para el desarrollo de aplicaciones web”. Ediciones Díaz de Santos Ltda.
- [7] Niño J. (2010), Aplicaciones web. EDITEX eBooks.
- [8] CMS, Open Source, <http://www.opensourcecms.com/>
- [9] Roger S. Pressman, (2005), Ingeniería del Software: Un Enfoque Práctico, 6ª Ed, McGraw-Hill
- [10] A. Ginige, S. Murusan, (2001), “Web Engineering: An Introduction”, IEEE Multimedia
- [11] Tramullas, J. (2013), Evaluación comparativa sobre CMS libres, <http://tramullas.com/evaluaciones-comparativas-sobre-cms-libres/>

- [12] Centro de Apoyo Tecnológico a Emprendedores de la Fund. Parque Científico y Tecnológico de Albacete, (2012), Estudio de los sistemas de gestión de contenidos web. Análisis de las mejores soluciones del mercado, <http://www.bilib.es/noticias/noticia/articulo/estudio-de-los-sistema-de-gestion-de-contenidos-w/>
- [13] The World Wide Web Consortium (W3C), <http://www.w3.org>
- [14] Plan 2014 para HTML5, (2013), HTML Working Group (W3C), <http://dev.w3.org/html5/decision-policy/html5-2014-plan.html>
- [15] W3C, WAI, Web Content Accessibility Guidelines (WCAG) 2.0, <http://www.w3.org/TR/WCAG/>
- [16] Comité ECMA, [www.ecma-international.org](http://www.ecma-international.org)
- [17] World Wide Web Technology Surveys (W3TECHS), <http://w3techs.com/>
- [18] PHP, <http://php.net/>
- [19] MySQL, <http://www.mysql.com/>
- [20] Merkel, Dirk (2010). Chapter 6: PHP Frameworks. Expert PHP 5 Tools. Packt Publishing.
- [21] Yii Framework, <http://www.yiiframework.com>
- [22] Apache Server, <http://www.apache.org/>
- [23] Aptana, <http://aptana.com/>
- [24] Netcraft, <http://news.netcraft.com/>
- [25] Eclipse, <http://www.eclipse.org/>
- [26] XAMPP, <http://www.apachefriends.org/es/xampp-windows.html>
- [27] Nils-Erik Frantzell, Install XAMPP for easy, integrated development, IBM,Developer Works, 30 Nov 2004
- [28] TeamLab, <http://www.teamlab.com/es/>
- [29] Google, <https://www.google.com/intl/es/chrome>
- [30] Quiñones D., “Alojamiento de Hosting” STRATO, 09/10/2012, <http://www.alojamientodehosting.net>
- [31] Durán Toro A., Bernárdez Jiménez A. Informe Técnico LSI-2000-10, 2002, Metodología para la Elicitación de Requisitos de Sistemas Software, <http://www.lsi.us.es/~informes/lsi-2000-10.pdf>.
- [32] Larman, C (2004). “UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado”. Prentice Hall, cap. 4, 5 y 7.
- [33] IEEE Recommended practice for software requirements specification. 1998. Artículo obtenido de la web del instituto de ingenieros eléctricos y electrónicos (Institute of Electrical and Electronics Engineers). <http://www.computer.org>
- [34] García S., (2012), Especificación de requisitos software con IEEE 830 , Godtic, <http://www.godtic.com/blog/2012/11/18/especificacion-de-requisitos-software-con-ieee-830-1998/>
- [35] Gause D.C. y G.M. Weinberg, (2011), Exploring Requirements: Quality before design. Ed: Dorset House.
- [36] Alarcón Caverro P., Yagüe Panadero A., (2002), Asignatura: Técnicas Avanzadas para el Desarrollo de Aplicaciones. Universidad Politécnica de Madrid, <http://www.oei.eui.upm.es/Asignaturas/BD/BD/docbd/tema/Arquitectura.pdf>
- [37] Arquitectura Modelo-Vista-Controlador. <http://javaweb.osmosislatina.com/curso/mvc.htm>
- [38] Pitt C. (2012) “PRO PHP MVC”, APRESS.



- [39] Unified Modeling Language (UML). <http://www.uml.org>
- [40] W.S. Humphrey: "A discipline for software engineering". Ed. Addison Wesley.1995.
- [41] A. Cuevas, (2003), Gestión del proceso software, Ed: Editorial Universitaria Ramón Areces
- [42] Casallas R., López N., Meneses R., (2012) Proyecto de construcción de software, Departamento de Sistemas y Computación, Facultad de Ingeniería, Universidad de los Andes  
<http://sistemas.uniandes.edu.co/~isis2603/dokuwiki/lib/exe/fetch.php?media=principal:isis2603-pruebasjuegogerencial.pdf>
- [43] Convención Internacional Sobre los Derechos de las Personas con Discapacidad - Artículo 1. Propósito. Pág. 4", 30 de marzo de 2007



# ANEXO I

## Catálogo de requisitos

Este anexo completa al apartado 4.4 de esta memoria. Se incluye la especificación de requisitos de la aplicación del sistema de gestión de contenidos objetivo de este proyecto. Éste recoge los requisitos de capacidad y de restricción para el CMS y establece el punto de partida para el análisis funcional del CMS.

Se ha seguido en gran medida la clasificación marcada por la norma IEEE, 1998 [33] sobre especificación de requisitos.

Los requisitos se han clasificado en los siguientes tipos:

- Requisitos de datos
- Requisitos de aplicación
- Requisitos de interfaz
- Requisitos de funcionalidades
- Requisitos de rendimiento
- Requisitos de seguridad

### ***1. Requisitos de datos***

Los requisitos de datos, también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información, responden a la pregunta de qué información debe almacenar y administrar la aplicación [34].

REQ.D.001	Información sobre Lenguajes
Datos	Clave primaria del idioma
	Visibilidad frente al usuario
	Nombre
	Abreviatura del idioma
	Texto alternativo
	Imagen

Tabla 1: Requisito de datos REQ.D.001: Información sobre Lenguajes

En requisitos de datos como el REQ.D.003 que se muestra a continuación la información se divide en dos tablas a la hora de grabarlo en la Base de Datos. Una de las tablas guarda la información referente al Teatro (Tabla de la BBDD “theater”) de la que no se pueden hacer traducciones, mientras que la otra guarda atributos referentes a los teatros susceptibles de traducción (Tabla de la BBDD “theaterinfo”). En los siguientes requisitos se asumirá que toda información sobre los elementos está almacenada en una sola tabla.

En apartados posteriores se explicara detalladamente la jerarquía de tablas y dependencia entre ellas.

REQ.D.002	Información sobre Teatros
Datos	Clave primaria del teatro
	Visibilidad frente al usuario
	Nombre
	Clave primaria de la ciudad
	Dirección
	Código postal
	Longitud
	Latitud
	Link
	Disponibilidad de Bucle magnético
	Acceso para personas con movilidad reducida
	Clave primaria de la traducción
	Clave primaria del teatro al que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción
	Historia del teatro

Tabla 2: Requisito de datos REQ.D.002: Información sobre Teatros

REQ.D.003	Información sobre Ciudades
Datos	Clave primaria de la ciudad
	Visibilidad frente al usuario
	Nombre

Tabla 3: Requisito de datos REQ.D.003: Información sobre Ciudades

REQ.D.004	Información sobre Temporadas
Datos	Clave primaria de la temporada
	Visibilidad frente al usuario
	Información sobre si la temporada esta activa
	Imagen
	Información sobre si pertenece a un festival
	Nombre
	Texto descriptivo
	Texto alternativo para la imagen
	Clave primaria de la traducción
	Clave primaria de la temporada a la que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción

Tabla 4: Requisito de datos REQ.D.004: Información sobre Temporadas

REQ.D.005	Información sobre Producciones
Datos	Clave primaria de la producción
	Nombre
	Nombre del autor
	Imagen banner
	Video
	Imagen
	Clave primaria de la información de la producción teatral
	Clave primaria de la producción a la que pertenece la traducción
	Resumen de sinopsis
	Sinopsis
	Texto alternativo de la imagen del poster
	Clave primaria del idioma al que pertenece la traducción
	Clave primaria del integrante del equipo de la producción
	Orden del elemento cuando se presenta
	Clave primaria de la traducción de la producción teatral a la que pertenece el integrante del equipo
	Puesto del integrante del equipo
	Nombre del integrante del equipo

Tabla 5: Requisito de datos REQ.D.005: Información sobre Producciones

REQ.D.006	Información sobre Performance
Datos	Clave primaria de la performance
	Visibilidad frente al usuario
	Clave primaria de la temporada
	Clave primaria del teatro
	Clave primaria de la producción
	Clave primaria de la traducción
	Clave primaria de la representación teatral
	Clave primaria del idioma al que pertenece la traducción
	Información sobre el subtitulado
	Información sobre la lengua de signos
	Información sobre la audio-descripción
	Información sobre el sonido de la sala
	Información sobre descuentos
	Información relacionada con la organización de grupos de usuarios

Tabla 6: Requisito de datos REQ.D.006: Información sobre Performance

REQ.D.007	Información sobre Portadas
Datos	Clave primaria de la portada
	Visibilidad frente al usuario
	Imagen
	Link
	Fecha a partir de la cual la portada es visible
	Fecha a partir de la cual la portada no es visible
	Clave primaria de la traducción
	Clave primaria del idioma al que pertenece la traducción
	Clave primaria de la portada a la que pertenece la traducción
	Traducción del título de la portada
	Traducción del texto de la portada
	Traducción del texto alternativo de la imagen de la portada

Tabla 7: Requisito de datos REQ.D.007: Información sobre Portadas

REQ.D.008	Información sobre Links
Datos	Clave primaria del enlace
	Visibilidad frente al usuario
	Imagen
	URL del enlace
	Tipo de enlace
	Orden del enlace cuando se presenta
	Clave primaria de la traducción
	Clave primaria del enlace al que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción
	Traducción del nombre del enlace
	Traducción del texto del enlace
	Traducción del texto alternativo de la imagen del enlace

Tabla 8: Requisito de datos REQ.D.008: Información sobre Links

REQ.D.009	Información sobre Noticias
Datos	Clave primaria de la noticia
	Visibilidad frente al usuario
	Fecha de publicación
	Clave primaria de la traducción
	Clave primaria de la noticia a la que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción
	Titular
	Resumen
	Texto

Tabla 9: Requisito de datos REQ.D.009: Información sobre Noticias

REQ.D.010	Información sobre Galerías
Datos	Clave primaria de la galería
	Visibilidad frente al usuario
	Fecha de publicación
	Clave primaria de la traducción
	Clave primaria de la galería a la que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción
	Nombre
	Texto

Tabla 10: Requisito de datos REQ.D.010: Información sobre Galerías

REQ.D.011	Información sobre Contenido
Datos	Clave primaria del contenido
	Visibilidad frente al usuario
	Nombre del archivo de la imagen en miniatura asociada al contenido
	URL que enlaza al contenido
	Clave primaria de la galería a la que pertenece el contenido
	Clave primaria del elemento
	Nombre del contenido
	Texto alternativo para la miniatura
	Tipo de contenido
	Clave primaria del contenido al que pertenece la traducción
	Clave primaria del idioma al que pertenece la traducción

Tabla 11: Requisito de datos REQ.D.011: Información sobre Contenido

REQ.D.012	Información sobre Sesión
Datos	Clave primaria de la fecha de la representación
	Visibilidad frente al usuario
	Clave primaria de la representación teatral
	Fecha de la representación

Tabla 12: Requisito de datos REQ.D.012: Información sobre Sesión

## 2. Requisitos de aplicación

Los requisitos de aplicación o sistema especifican el entorno tecnológico y los componentes hardware y software necesarios para la implantación y explotación de la aplicación.

REQ.A.001	Plataforma Hardware
Descripción	La plataforma hardware que soporte la aplicación debe ser un PC compatible capaz de ejecutar el S.O. Windows.

Tabla 13: Requisito de aplicación REQ.A.001: Plataforma Hardware

REQ.A.002	Consistencia en la Base de datos
Descripción	<p>Todo aquello que el administrador del gestor de contenidos modifique, cree o elimine debe guardar siempre el objetivo de consistencia.</p> <p>Se trata de que la base de datos no se vea afectada a la hora de recuperar datos, es decir, no se pueden eliminar las dependencias entre tablas de la base de datos si esto va a crear una incongruencia en el código y a su vez un error en la página.</p>

Tabla 14: Requisito de aplicación REQ.A.002: Consistencia en la Base de datos



## 2.1 Requisitos de aplicación impuestos

Los requisitos de restricción engloban a las limitaciones impuestas por la empresa sobre la forma en la que se resuelve un problema o alcanza un objetivo.

REQ.A.003	Servicio de hosting STRATO
Descripción	<p>Forma parte de uno de los requisitos impuestos por el cliente.</p> <p>La plataforma de alojamiento web (hosting) ha de asegurar una gestión segura del CMS. Además ha de ser compatible con lenguajes de programación como PHP y MySQL.</p> <p>Strato ha optimizado la nueva plataforma para conseguir unos tiempos de ejecución de los scripts PHP más cortos y un alto rendimiento respecto a las bases de datos MySQL. Esto incluye servidores configurados específicamente y modificaciones en el servidor web Apache, así como adaptaciones en todos los entornos script. Strato ha unificado toda esta nueva tecnología en una única plataforma para todos los paquetes de hosting.</p>

Tabla 15: Requisito de aplicación REQ.A.003: Servicio de hosting STRATO

REQ.A.004	Lenguaje PHP
Descripción	<p>La utilización de lenguaje de programación en PHP forma parte de uno de los requisitos impuestos por el cliente.</p> <p>Se trata de un lenguaje de programación muy orientado al desarrollo web. Tiene una curva de aprendizaje muy baja y se alcanza una gran productividad con poco esfuerzo. Es un lenguaje de script interpretado lo que lo hace algo más lento en ejecución que java.</p>

Tabla 16: Requisito de aplicación REQ.A.004: Lenguaje PHP

REQ.A.005	Base de datos en MySQL
Descripción	<p>MySQL como gestor de base de datos es uno de los requisitos impuestos por el cliente.</p> <p>MySQL es un sistema gestor de administración para bases de datos relacionales. Destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.</p>

Tabla 17: Requisito de aplicación REQ.A.005: Base de datos en MySQL

REQ.A.006	Plataforma de trabajo Yii Framework
Descripción	<p>La utilización de este framework era obligatoria debido a que la página web en la que se basa el CMS está desarrollada en este framework de estilo MVC (Modelo-Vista-Controlador).</p> <p>Yii es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo.</p>

Tabla 18: Requisito de aplicación REQ.A.006: Plataforma de trabajo Yii Framework

<b>REQ.A.007</b>	<b>Entorno de desarrollo Aptana Studio</b>
<b>Descripción</b>	El entorno de desarrollo integrado ha sido impuesto por la empresa que ya utilizaba esta plataforma. Aptana Studio 3 es una herramienta que nos permite desarrollar y probar aplicaciones Web en un único entorno. Tiene soporte para las especificaciones de última tecnología para los navegadores, como HTML5, CSS3, Javascript, Ruby, Rails, PHP y Python.

Tabla 19: Requisito de aplicación REQ.A.007: Entorno de desarrollo Aptana Studio

### 3. Requisitos de la Interfaz

Los requisitos de la interfaz, también llamados requisitos de interacción de usuario, responden a la pregunta de cómo va a interactuar el usuario con la aplicación.

<b>REQ.I.001</b>	<b>Perfil de Administrador</b>
<b>Descripción</b>	El único perfil como usuario será el de administración que tras un Login dará acceso a las pertinentes funcionalidades de la aplicación.

Tabla 20: Requisito de interfaz REQ.I.001: Perfil de Administrador

<b>REQ.I.002</b>	<b>Inicio de sesión</b>
<b>Descripción</b>	Todos los usuarios de la aplicación deben realizar un inicio de sesión previo a usar cualquier otra funcionalidad. El inicio de sesión de cada usuario debe autenticarse mediante usuario y contraseña.

Tabla 21: Requisito de interfaz REQ.I.002: Inicio de sesión

<b>REQ.I.003</b>	<b>Cierre de sesión</b>
<b>Descripción</b>	Una vez iniciada una sesión por un usuario, la sesión podrá cerrarse por la acción del usuario a través de la opción de "logout"

Tabla 22: Requisito de interfaz REQ.I.003: Cierre de sesión

<b>REQ.I.004</b>	<b>Usabilidad de controles y vistas</b>
<b>Descripción</b>	La aplicación podrá usarse al menos con los tres navegadores Web más extendidos: Microsoft Internet Explorer, Google Chrome y Firefox. No se requerirá la instalación de ningún software adicional por parte de los usuarios.

Tabla 23: Requisito de interfaz REQ.I.004: Usabilidad de controles y vistas

### 4. Requisitos de Funcionalidades

Los requisitos funcionales especifican las funciones que ha de desempeñar la aplicación [35].

Se subdividen en dos grupos diferenciados:

1. **Contenido de Programación:** son imprescindibles y como tales no pueden ser eliminados.
2. **Contenido Adicional:** pueden verse modificados o sacados de la base de datos en cualquier momento si el administrador “base” lo permite.

<b>REQ.F.001</b>	<b>Gestión de contenido de programación</b>
<b>Descripción</b>	Creación o edición de contenido referente a la programación de obras de teatro.
<b>Comentarios</b>	Entre el tipo de contenido posible está: Ciudad, Teatro, Temporada, Performance, Sesión y Producción.

Tabla 24: Requisito de funcionalidad REQ.F.001: Gestión de contenido de programación

<b>REQ.F.002</b>	<b>Gestión de contenido adicional</b>
<b>Descripción</b>	Creación, edición o eliminación de contenido no referente a la programación de obras de teatro.
<b>Comentarios</b>	Entre el tipo de contenido posible está: Link, Galería, Noticias, Portadas y Contenidos.

Tabla 25: Requisito de funcionalidad REQ.F.002: Gestión de contenido adicional

<b>REQ.F.003</b>	<b>Gestión de Idiomas</b>
<b>Descripción</b>	Creación, edición o eliminación de idiomas en la base de datos
<b>Comentarios</b>	Esto dará la posibilidad de generar distintas traducciones en el contenido.

Tabla 26: Requisito de funcionalidad REQ.F.003: Gestión de Idiomas

#### ***4.1 Requisitos de Funcionalidades dependientes del REQ.F.001***

<b>REQ.F.001.1</b>	<b>Añadir nueva Ciudad</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Ciudad

Tabla 27: Requisito de funcionalidad REQ.F.001.1: Añadir nueva Ciudad

<b>REQ.F.001.2</b>	<b>Editar Ciudad</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Ciudad seleccionado

Tabla 28: Requisito de funcionalidad REQ.F.001.2: Editar Ciudad

<b>REQ.F.001.3</b>	<b>Añadir nuevo Teatro</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Teatro

Tabla 29: Requisito de funcionalidad REQ.F.001.3: Añadir nuevo Teatro

<b>REQ.F.001.4</b>	<b>Editar Teatro</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Teatro seleccionado

Tabla 30: Requisito de funcionalidad REQ.F.001.4: Editar Teatro

<b>REQ.F.001.5</b>	<b>Añadir nueva Temporada</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Temporada

Tabla 31: Requisito de funcionalidad REQ.F.001.5: Añadir nueva Temporada

<b>REQ.F.001.6</b>	<b>Editar Temporada</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Temporada seleccionado

Tabla 32: Requisito de funcionalidad REQ.F.001.6: Editar Temporada

<b>REQ.F.001.7</b>	<b>Añadir nueva Producción</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Producción

Tabla 33: Requisito de funcionalidad REQ.F.001.7: Añadir nueva Producción

<b>REQ.F.001.8</b>	<b>Editar Producción</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Producción seleccionado

Tabla 34: Requisito de funcionalidad REQ.F.001.8: Editar Producción

<b>REQ.F.001.9</b>	<b>Añadir nueva Performance</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Performance

Tabla 35: Requisito de funcionalidad REQ.F.001.9: Añadir nueva Performance

<b>REQ.F.001.10</b>	<b>Editar Performance</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Performance seleccionado

Tabla 36: Requisito de funcionalidad REQ.F.001.10: Editar Performance

<b>REQ.F.001.11</b>	<b>Añadir nueva Sesión</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Sesión

Tabla 37: Requisito de funcionalidad REQ.F.001.11: Añadir nueva Sesión

<b>REQ.F.001.12</b>	<b>Editar Sesión</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Sesión seleccionado

Tabla 38: Requisito de funcionalidad REQ.F.001.12: Editar Sesión

#### ***4.2 Requisitos de Funcionalidades dependientes del REQ.F.002***

<b>REQ.F.002.1</b>	<b>Añadir nueva Galería</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Galería

Tabla 39: Requisito de funcionalidad REQ.F.002.1: Añadir nueva Galería

<b>REQ.F.002.2</b>	<b>Editar Galería</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Galería seleccionado

Tabla 40: Requisito de funcionalidad REQ.F.002.2: Editar Galería

<b>REQ.F.002.3</b>	<b>Eliminar Galería</b>
<b>Descripción</b>	Eliminar de la base de datos el elemento de tipo Galería seleccionado

Tabla 41: Requisito de funcionalidad REQ.F.002.3: Eliminar Galería

<b>REQ.F.002.4</b>	<b>Añadir nueva Noticia</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Noticias

Tabla 42: Requisito de funcionalidad REQ.F.002.4: Añadir nueva Noticia

<b>REQ.F.002.5</b>	<b>Editar Noticias</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Noticias seleccionado

Tabla 43: Requisito de funcionalidad REQ.F.002.5: Editar Noticias

<b>REQ.F.002.6</b>	<b>Eliminar Noticias</b>
<b>Descripción</b>	Eliminar de la base de datos el elemento de tipo Noticias seleccionado

Tabla 44: Requisito de funcionalidad REQ.F.002.6: Eliminar Noticias

<b>REQ.F.002.7</b>	<b>Añadir nuevo Link</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Link

Tabla 45: Requisito de funcionalidad REQ.F.002.7: Añadir nuevo Link

<b>REQ.F.002.8</b>	<b>Editar Link</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Link seleccionado

Tabla 46: Requisito de funcionalidad REQ.F.002.8: Editar Link

<b>REQ.F.002.9</b>	<b>Eliminar Link</b>
<b>Descripción</b>	Eliminar de la base de datos el elemento de tipo Link seleccionado

Tabla 47: Requisito de funcionalidad REQ.F.002.9: Eliminar Link

<b>REQ.F.002.10</b>	<b>Añadir nueva Portada</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Portada

Tabla 48: Requisito de funcionalidad REQ.F.002.10: Añadir nueva Portada

<b>REQ.F.002.11</b>	<b>Editar Portada</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Portada seleccionado

Tabla 49: Requisito de funcionalidad REQ.F.002.11: Añadir nueva Portada

<b>REQ.F.002.12</b>	<b>Eliminar Portada</b>
<b>Descripción</b>	Eliminar de la base de datos el elemento de tipo Portada seleccionado

Tabla 50: Requisito de funcionalidad REQ.F.002.12: Eliminar Portada

<b>REQ.F.002.13</b>	<b>Añadir nuevo Contenido</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Contenido

Tabla 51: Requisito de funcionalidad REQ.F.002.13: Añadir nuevo Contenido

<b>REQ.F.002.14</b>	<b>Editar Contenido</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Contenido seleccionado

Tabla 52: Requisito de funcionalidad REQ.F.002.14: Editar Contenido

<b>REQ.F.002.15</b>	<b>Eliminar Contenido</b>
<b>Descripción</b>	Eliminar de la base de datos el elemento de tipo Contenido seleccionado

Tabla 53: Requisito de funcionalidad REQ.F.002.15: Eliminar Contenido

### ***4.3 Requisitos de Funcionalidades dependientes del REQ.F.003***

<b>REQ.F.003.1</b>	<b>Añadir nuevo Lenguaje</b>
<b>Descripción</b>	Creación en la base de datos de un nuevo elemento del tipo Lenguaje

Tabla 54: Requisito de funcionalidad REQ.F.003.1: Añadir nuevo Lenguaje

<b>REQ.F.003.2</b>	<b>Editar Lenguaje</b>
<b>Descripción</b>	Modificar atributos en la base de datos del elemento Lenguaje seleccionado

Tabla 55: Requisito de funcionalidad REQ.F.003.2: Editar Lenguaje

## ***5. Requisitos de Rendimiento***

Los requisitos de rendimiento especifican los recursos máximos que puede consumir la aplicación [36].

<b>REQ.R.001</b>	<b>Tiempo de Respuesta</b>
<b>Descripción</b>	El tiempo de respuesta de la interfaz Web de la aplicación debe ser inferior a 1 segundo (sin tener en cuenta retardos de red).

Tabla 56: Requisito de rendimiento REQ.R.001: Tiempo de Respuesta

<b>REQ.R.002</b>	<b>Carga computacional</b>
<b>Descripción</b>	El conjunto del software necesario para la prestación del servicio debe poder ejecutarse en un solo PC compatible actual con 500MB de memoria principal.

Tabla 57: Requisito de rendimiento REQ.R.002: Carga computacional

## ***6. Requisitos de Seguridad***

Los requisitos de seguridad especifican las medidas de seguridad implementadas en la aplicación.

<b>REQ.S.001</b>	<b>Almacenamiento de contraseñas</b>
<b>Descripción</b>	Las contraseñas serán almacenadas en la base de datos en un formato cifrado.

Tabla 58: Requisito de seguridad REQ.S.001: Almacenamiento de contraseñas

<b>REQ.S.002</b>	<b>Seguridad del servidor web</b>
<b>Descripción</b>	La seguridad del proveedor debe facilitarnos un Backup como copia de seguridad frente a posibles amenazas además de una protección frente a virus para que no solo la página web, sino también sus visitantes estén protegidos.

Tabla 59: Requisito de seguridad REQ.S.002: Seguridad del servidor web

# **ANEXO II**

## **Casos de uso**

Este Anexo completa al apartado 4.5 de esta memoria. En este anexo se incluye la especificación de casos de uso de la aplicación del sistema de gestión de contenidos objetivo de este proyecto. Se describen en un principio los casos de uso básicos que la aplicación ha de cumplir para posteriormente describir los casos de uso que se heredan de los principales.

Los casos de uso están definidos de manera que se puedan relacionar con requisitos exigidos de la aplicación además de poseer una secuencia de los pasos a seguir para poder obtener el resultado deseado.



## 1. Casos de uso para principales de la aplicación.

<b>C.U.001</b>	<b>Autenticación de usuario</b>	
<b>Descripción</b>	Realiza el proceso de verificación de los usuarios antes de entrar en la aplicación. El proceso de login controla el acceso de los usuarios al sistema, de tal forma que solo aquellos usuarios que validan correctamente su login y su contraseña pueden acceder a la página principal de la aplicación. La función principal de esta operación es evitar el acceso de usuarios no deseados.	
<b>Requisito Relacionado</b>	REQ.I.001 - Perfil de Administrador REQ.I.002 – Inicio de sesión REQ.S.001 – Almacenamiento de contraseñas	
<b>Precondición</b>		
<b>Postcondición</b>	El usuario pasa a estar conectado y se encuentra en la página principal de la aplicación.	
<b>Observaciones</b>	En caso de no coincidir la información con las credenciales se denegará el acceso al gestor de contenidos.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El presente caso de uso se inicia cuando el usuario intenta acceder al gestor de contenidos. Se mostrará un formulario donde deberá introducir el usuario y contraseña.
	2	El usuario introduce sus datos y acepta el formulario.
	3	El sistema valida al usuario contra la base de datos y le da permisos de administrador.
	4	Se muestra un mensaje de bienvenida con instrucciones sobre la utilización de la aplicación web
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	Si el usuario no introduce correctamente alguno de los campos de validación (Usuario o Contraseña) se le devolverá de nuevo el formulario especificando el problema.

Tabla 1: C.U.001: Autenticación de usuario

<b>C.U.002</b>	<b>Gestión de contenido de programación</b>	
<b>Descripción</b>	El usuario podrá acceder al contenido de programación en el menú izquierdo pudiendo así crear o modificar elementos en la base de datos como usuario administrador.	
<b>Requisito Relacionado</b>	REQ.A.002 – Consistencia en la Base de Datos REQ.F.001 - Gestión de contenido de programación REQ.D.002 – Información sobre Teatros REQ.D.003 – Información sobre Ciudades REQ.D.004 – Información sobre Temporadas REQ.D.005 – Información sobre Producción REQ.D.006 – Información sobre Performance REQ.D.012 – Información sobre Sesión	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Se modificará la base de datos con los elementos con los que el administrador haya estado trabajando.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa uno de los elementos del menú izquierdo.
	2	Se muestra una tabla con todos los elementos del tipo seleccionado que existen en la base de datos.
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de que la lista no contenga elementos no se mostrará la tabla con los elementos del tipo de elemento seleccionado.

Tabla 2: C.U.002: Gestión de contenido de programación

<b>C.U.013</b>	<b>Gestión de contenido adicional</b>	
<b>Descripción</b>	El usuario podrá acceder al contenido adicional en el menú izquierdo pudiendo así crear, eliminar o modificar elementos en la base de datos como usuario administrador.	
<b>Requisito Relacionado</b>	REQ.A.002 – Consistencia en la Base de Datos REQ.F.002 - Gestión de contenido adicional  REQ.D.007 – Información sobre Portadas REQ.D.008 – Información sobre Links REQ.D.009 – Información sobre Noticias REQ.D.010 – Información sobre Galerías REQ.D.011 – Información sobre Contenido	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Se modificará la base de datos con los elementos con los que el administrador haya estado trabajando.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa uno de los elementos del menú izquierdo.
	2	Se muestra una tabla con todos los elementos del tipo seleccionado que existen en la base de datos.
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de que la lista no contenga elementos no se mostrará la tabla con los elementos del tipo de elemento seleccionado.

Tabla 3: C.U.013: Gestión de contenido adicional

<b>C.U.026</b>	<b>Gestión de idiomas</b>	
<b>Descripción</b>	El usuario podrá acceder al contenido del lenguaje en el menú izquierdo pudiendo así crear o modificar algún idioma en la base de datos	
<b>Requisito Relacionado</b>	REQ.A.002 – Consistencia en la Base de Datos REQ.F.003 - Gestión de idiomas  REQ.D.001 – Información sobre Lenguaje	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Se modificará la base de datos con los elementos con los que el administrador haya estado trabajando.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa uno de los elementos del menú izquierdo.
	2	Se muestra una tabla con todos los elementos del tipo seleccionado que existen en la base de datos.
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de que la lista no contenga elementos no se mostrará la tabla con los elementos del tipo de elemento seleccionado.

Tabla 4: C.U.026: Gestión de idiomas

<b>C.U.029</b>	<b>Cierre de sesión</b>	
<b>Descripción</b>	El sistema cerrará la sesión del usuario.	
<b>Requisito Relacionado</b>	REQ.I.004 – Cierre de sesión	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Se cerrará la sesión del usuario.	
<b>Observaciones</b>	Se devolverá al usuario a la página principal de Teatro Accesible.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa “Logout” de la parte derecha superior
	2	El sistema cierra la sesión del usuario en la aplicación
	3	Se muestra la página principal de Teatro Accesible
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	Si se produce un error en el cierre de sesión se muestra la página de error.

Tabla 5: C.U.029: Cierre de sesión

### ***3. Casos de uso para la Gestión de contenido de Programación.***

<b>C.U.003</b>	<b>Añadir Ciudad</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Ciudad. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.003 – Información sobre Ciudades REQ.F.001 – Gestión de contenido de Programación REQ.F.001.1 – Añadir nueva Ciudad	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Ciudades
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 6: C.U.003: Añadir Ciudad

<b>C.U.004</b>	<b>Editar Ciudad</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.003 – Información sobre Ciudades REQ.F.001 – Gestión de contenido de Programación REQ.F.001.2 – Editar Ciudad	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Ciudades
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	6	El sistema muestra los detalles del elemento guardado
	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 7: C.U.004: Editar Ciudad

<b>C.U.005</b>	<b>Añadir Teatro</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de un nuevo Teatro. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.002 – Información sobre Teatros REQ.F.001 – Gestión de contenido de Programación REQ.F.001.3 – Añadir nuevo Teatro	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Teatro
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 8: C.U.005: Añadir Teatro

<b>C.U.006</b>	<b>Editar Teatro</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.002 – Información sobre Teatros REQ.F.001 – Gestión de contenido de Programación REQ.F.001.4 – Editar Teatro	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Teatro
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 9: C.U.006: Editar Teatro

<b>C.U.007</b>	<b>Añadir Temporada</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Temporada. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.004 – Información sobre Temporada REQ.F.001 – Gestión de contenido de Programación REQ.F.001.5 – Añadir nueva Temporada	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Temporada
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 10: C.U.007: Añadir Temporada



<b>C.U.008</b>	<b>Editar Temporada</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.004 – Información sobre Temporada REQ.F.001 – Gestión de contenido de Programación REQ.F.001.6 – Editar Temporada	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Temporada
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla "Editar"
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa "Guardar"
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 11: C.U.008: Editar Temporada

<b>C.U.009</b>	<b>Añadir Producción</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Producción. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.005 – Información sobre Producciones REQ.F.001 – Gestión de contenido de Programación REQ.F.001.7 – Añadir nueva Producción	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Producción
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 12: C.U.009: Añadir Producción

<b>C.U.010</b>	<b>Editar Producción</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.005 – Información sobre Producciones REQ.F.001 – Gestión de contenido de Programación REQ.F.001.8 – Editar Producción	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Producción
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	6	El sistema muestra los detalles del elemento guardado
	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 13: C.U.010: Editar Producción

<b>C.U.011</b>	<b>Añadir Performance</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Performance. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.002 – Información sobre Teatros REQ.D.004 – Información sobre Temporada REQ.D.005– Información sobre Producciones REQ.D.006 – Información sobre Performance REQ.F.001 – Gestión de contenido de Programación REQ.F.001.9 – Añadir nueva Performance	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Performance
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 14: C.U.011: Añadir Performance

<b>C.U.012</b>	<b>Editar Performance</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.002 – Información sobre Teatros REQ.D.004 – Información sobre Temporada REQ.D.005– Información sobre Producciones REQ.D.006 – Información sobre Performance REQ.F.001 – Gestión de contenido de Programación REQ.F.001.10 – Editar Performance	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Performance
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 15: C.U.012: Editar Performance

<b>C.U.012.1</b>	<b>Añadir Sesión</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Sesión. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.012 – Información sobre Sesión REQ.D.006 – Información sobre Performance REQ.F.001 – Gestión de contenido adicional REQ.F.001.10 - Editar Performance REQ.F.001.11 – Añadir nueva Sesión	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Performance
	2	El usuario selecciona uno de los elementos de la lista mostrada
	3	El usuario pulsa “Editar”
	4	El usuario pulsa el botón “Añadir sesión” en la parte inferior de la página
	5	El sistema despliega un nuevo formulario para rellenar sesiones
	6	El usuario rellena todos los campos del formulario y pulsa “Guardar”
	7	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	8	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 16: C.U.012.1: Añadir Sesión

<b>C.U.012.2</b>	<b>Editar Sesión</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.012 – Información sobre Sesión REQ.D.006 – Información sobre Performance REQ.F.001 – Gestión de contenido adicional REQ.F.001.10 - Editar Performance REQ.F.001.12 – Editar Sesión	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Performance
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 17: C.U.012.2: Editar Sesión

#### 4. Casos de uso para la Gestión de contenido adicional

<b>C.U.014</b>	<b>Añadir Link</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de un nuevo Link. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.008 – Información sobre Links REQ.F.002 – Gestión de contenido adicional REQ.F.002.10 – Añadir nuevo Link	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Links
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 18: C.U.014: Añadir Link



<b>C.U.015</b>	<b>Editar Link</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.008 – Información sobre Links REQ.F.002 – Gestión de contenido adicional REQ.F.002.11 – Editar Link	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Links
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 19: C.U.015: Editar Link

<b>C.U.016</b>	<b>Eliminar Link</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a eliminar tendrá que confirmar dos veces que desea borrar el elemento seleccionado para que así éste quede borrado de la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.008 – Información sobre Links REQ.F.002 – Gestión de contenido adicional REQ.F.002.12 – Eliminar Link	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán eliminados los datos del elemento de la tabla correspondiente en la base de datos. Se le mostrará al usuario de nuevo la lista actualizada.	
<b>Observaciones</b>	Por motivos de seguridad se realiza una doble comprobación a través de dos alertas emergentes que el usuario ha de confirmar.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Links
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Eliminar”
	3	El sistema muestra una alerta emergente pidiendo confirmación
	4	El usuario acepta la eliminación del elemento
	5	El sistema vuelve a comprobar que se está conforme con la eliminación del elemento avisando de las posibles consecuencias
	6	El usuario acepta la eliminación del elemento
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	6	Si se produce un error al eliminar los datos se envía a la página de error

Tabla 20: C.U.016: Eliminar Link

<b>C.U.020</b>	<b>Añadir Noticia</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Noticia. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Noticias REQ.F.002 – Gestión de contenido adicional REQ.F.002.7 – Añadir nueva Noticia	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Noticias
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	6	El sistema muestra los detalles del elemento guardado
	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 21: C.U.020: Añadir Noticia

<b>C.U.021</b>	<b>Editar Noticia</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Noticias REQ.F.002 – Gestión de contenido adicional REQ.F.002.8 – Editar Noticia	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Noticias
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 22: C.U.021: Editar Noticia

<b>C.U.022</b>	<b>Eliminar Noticia</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a eliminar tendrá que confirmar dos veces que desea borrar el elemento seleccionado para que así éste quede borrado de la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Noticias REQ.F.002 – Gestión de contenido adicional REQ.F.002.9 – Eliminar Noticia	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán eliminados los datos del elemento de la tabla correspondiente en la base de datos. Se le mostrará al usuario de nuevo la lista actualizada.	
<b>Observaciones</b>	Por motivos de seguridad se realiza una doble comprobación a través de dos alertas emergentes que el usuario ha de confirmar.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Noticias
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Eliminar”
	3	El sistema muestra una alerta emergente pidiendo confirmación
	4	El usuario acepta la eliminación del elemento
	5	El sistema vuelve a comprobar que se está conforme con la eliminación del elemento avisando de las posibles consecuencias
	6	El usuario acepta la eliminación del elemento
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	6	Si se produce un error al eliminar los datos se envía a la página de error

Tabla 23: C.U.022: Eliminar Noticia

<b>C.U.023</b>	<b>Añadir Portada</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Portada. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Noticias REQ.F.002 – Gestión de contenido adicional REQ.F.002.7 – Añadir nueva Portada	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Portadas
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 24: C.U.023: Añadir Portada

<b>C.U.0024</b>	<b>Editar Portada</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Portada REQ.F.002 – Gestión de contenido adicional REQ.F.002.14 – Editar Portada	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Portadas
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 25: C.U.024: Editar Portada

<b>C.U.025</b>	<b>Eliminar Portada</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a eliminar tendrá que confirmar dos veces que desea borrar el elemento seleccionado para que así éste quede borrado de la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.009 – Información sobre Portada REQ.F.002 – Gestión de contenido adicional REQ.F.002.15 – Eliminar Portada	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán eliminados los datos del elemento de la tabla correspondiente en la base de datos. Se le mostrará al usuario de nuevo la lista actualizada.	
<b>Observaciones</b>	Por motivos de seguridad se realiza una doble comprobación a través de dos alertas emergentes que el usuario ha de confirmar.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Portadas
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Eliminar”
	3	El sistema muestra una alerta emergente pidiendo confirmación
	4	El usuario acepta la eliminación del elemento
	5	El sistema vuelve a comprobar que se está conforme con la eliminación del elemento avisando de las posibles consecuencias
	6	El usuario acepta la eliminación del elemento
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	6	Si se produce un error al eliminar los datos se envía a la página de error

Tabla 26: C.U.025: Eliminar Portada



<b>C.U.017</b>	<b>Añadir Galería</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de una nueva Galería. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.010 – Información sobre Galerías REQ.F.002 – Gestión de contenido adicional REQ.F.002.4 – Añadir nueva Galería	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	6	El sistema muestra los detalles del elemento guardado
	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 27: C.U.017: Añadir Galería

<b>C.U.018</b>	<b>Editar Galería</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.010 – Información sobre Galerías REQ.F.002 – Gestión de contenido adicional REQ.F.002.5 – Editar Galería	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 28: C.U.018: Editar Galería

<b>C.U.019</b>	<b>Eliminar Galería</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a eliminar tendrá que confirmar dos veces que desea borrar el elemento seleccionado para que así éste quede borrado de la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.010 – Información sobre Galerías REQ.F.002 – Gestión de contenido adicional REQ.F.002.6 – Eliminar Galería	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán eliminados los datos del elemento de la tabla correspondiente en la base de datos. Se le mostrará al usuario de nuevo la lista actualizada.	
<b>Observaciones</b>	Por motivos de seguridad se realiza una doble comprobación a través de dos alertas emergentes que el usuario ha de confirmar.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Eliminar”
	3	El sistema muestra una alerta emergente pidiendo confirmación
	4	El usuario acepta la eliminación del elemento
	5	El sistema vuelve a comprobar que se está conforme con la eliminación del elemento avisando de las posibles consecuencias
	6	El usuario acepta la eliminación del elemento
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	6	Si se produce un error al eliminar los datos se envía a la página de error

Tabla 29: C.U.019: Eliminar Galería

<b>C.U.018.1</b>	<b>Añadir Contenido</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de un nuevo Contenido. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.011 – Información sobre Contenido REQ.D.010 – Información sobre Galería REQ.F.002 – Gestión de contenido de Programación REQ.F.002.5 - Editar Galería REQ.F.002.16 – Añadir nuevo Contenido	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario selecciona uno de los elementos de la lista mostrada
	3	El usuario pulsa “Editar”
	4	El usuario pulsa el botón “Añadir Contenido” en la parte inferior de la página
	5	El sistema despliega un nuevo formulario por rellenar
	6	El usuario rellena todos los campos del formulario y pulsa “Guardar”
	7	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	8	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 30: C.U.018.1: Añadir Contenido

<b>C.U.018.2</b>	<b>Editar Contenido</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.011 – Información sobre Contenido REQ.D.010 – Información sobre Galería REQ.F.002 – Gestión de contenido de Programación REQ.F.002.5 - Editar Galería REQ.F.002.17 – Editar Contenido	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 31: C.U.018.2: Editar Contenido

<b>C.U.018.3</b>	<b>Eliminar Contenido</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a eliminar tendrá que confirmar dos veces que desea borrar el elemento seleccionado para que así éste quede borrado de la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.D.010 – Información sobre Galerías REQ.F.002 – Gestión de contenido adicional REQ.F.002.6 – Eliminar Galería REQ.F.002.18 – Eliminar Contenido	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán eliminados los datos del elemento de la tabla correspondiente en la base de datos. Se le mostrará al usuario de nuevo la lista actualizada.	
<b>Observaciones</b>	Por motivos de seguridad se realiza una doble comprobación a través de dos alertas emergentes que el usuario ha de confirmar.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Galería
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Eliminar”
	3	El sistema muestra una alerta emergente pidiendo confirmación
	4	El usuario acepta la eliminación del elemento
	5	El sistema vuelve a comprobar que se está conforme con la eliminación del elemento avisando de las posibles consecuencias
	6	El usuario acepta la eliminación del elemento
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	6	Si se produce un error al eliminar los datos se envía a la página de error

Tabla 32: C.U.018.3: Eliminar Contenido

## 5. Casos de uso para la Gestión de idiomas

<b>C.U.027</b>	<b>Añadir Lenguaje</b>	
<b>Descripción</b>	El usuario introducirá toda la información que el formulario le exija para la creación de un nuevo Lenguaje. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.F.003 – Gestión de idiomas REQ.F.003.1 – Añadir nuevo Lenguaje	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Idiomas
	2	El usuario pulsa en el menú superior de la tabla “Crear”
	3	El sistema muestra un formulario con los campos a rellenar
	4	El usuario rellena todos los campos y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 33: C.U.027: Añadir Lenguaje

<b>C.U.028</b>	<b>Editar Lenguaje</b>	
<b>Descripción</b>	El usuario tras seleccionar el elemento a editar modificará toda la información que desee a través del formulario que le mostrará la información actual del elemento. Todos los datos se almacenarán en la base de datos.	
<b>Requisito Relacionado</b>	REQ.D.001 – Información sobre Lenguajes REQ.F.003 – Gestión de idiomas REQ.F.003.2 – Editar Lenguaje	
<b>Precondición</b>	El usuario debe estar validado.	
<b>Postcondición</b>	Serán almacenados los datos suministrados por el usuario en la tabla correspondiente de la base de datos.	
<b>Observaciones</b>	En caso de que el tipo de dato introducido no sea correcto, la aplicación coloreará de rojo aquel campo que no haya sido introducido correctamente.	
<b>Secuencia de pasos</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario pulsa en el menú izquierdo Idiomas
	2	El usuario selecciona un elemento de la lista y pulsa en el menú superior de la tabla “Editar”
	3	El sistema muestra un formulario con los campos rellenos
	4	El usuario rellena de nuevo todos los campos que quiera modificar y pulsa “Guardar”
	5	El sistema almacena todos los atributos del elemento nuevo en la tabla correspondiente de la base de datos
	6	El sistema muestra los detalles del elemento guardado
<b>Secuencia de excepción</b>	<b>Paso</b>	<b>Excepción</b>
	2	En caso de no tener ningún elemento seleccionado saltará una alerta indicando este evento.
	3	En caso de no rellenar algún campo requerido o rellenarlo de manera incorrecta se muestran estos campos en rojo indicando que deben rellenarse obligatoriamente o con el formato correcto.
	4	Si se produce un error al almacenar los datos se envía a la página de error

Tabla 34: C.U.028: Editar Lenguaje



# **ANEXO III**

## **Catálogo de Pruebas**

En este anexo se termina de completar el apartado 6.4 del documento de la memoria. Tras la introducción sobre el tipo de pruebas que se realizan al CMS podemos ver en este anexo la especificación de cada una de las pruebas que se han realizado dándonos una visión global de la validación de la aplicación desarrollada y así confirmar que existe un correcto funcionamiento del CMS.

Prueba	P.001	Caso de Uso			C.U.001
Objetivo	Verificar que si se introduce un usuario y contraseña correctos se acceda a la administración de la aplicación.				
Criticidad	Alta	X	Media		Baja
Punto de medición	Desde que se hace clic en el botón “Login” hasta que se devuelve la página principal de la aplicación.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Verificar que aparece el menú de opciones en la página y el mensaje de bienvenida.			Formulario de autenticación de usuario indicando los campos erróneos o incompletos o si se produce algún error inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Usuario	admin		
		Contraseña	12345		
Comprobación	Usuario incorrecto	Usuario	admin	newuser	
		Contraseña	12345	12345	
Comprobación	Contraseña incorrecta	Usuario	admin	admin	
		Contraseña	12345	54321	

Tabla 1: P.001-Verificar el acceso a la aplicación

Prueba	P.002		Caso de Uso			C.U.003
Objetivo	Comprobar que un elemento del tipo Ciudad se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo	Ejecución			
Realizada	Mensaje		Correcta		Incorrecta	
Obligatorio	[Campo] es obligatorio	Nombre	Madrid			
		Visible	On/Off (*)			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Madrid		{x=100}	

Tabla 2: P.002-Comprobación de la correcta creación del elemento ciudad

Prueba	P.003		Caso de Uso		C.U.005
Objetivo	Comprobar que un elemento del tipo Teatro se crea correctamente				
Criticidad	Alta	X	Media		Baja
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Nombre	Teatro C.M.C		
		Ciudad	Madrid		
		Dirección	Avd/Carlos III		
		Latitud	40.411320		
		Longitud	-3.708740		
		Historia	La historia		
		Banner	t.jpg		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Teatro C.M.C	{x =100}	
		Código Postal	28024	{x =5}	
		Dirección	Avd/Carlos III	{x =100}	
		Link	www.link.es	{x =150}	
		Historia	La historia	{x =2000}	
		Banner	t.jpg	{x =40}	
Numérico	[Campo] debe ser numérico	Código Postal	28025	Tres	
		Latitud	40.411320	Cuatro	
		Longitud	-3.708740	Cinco	
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner	t.jpg	t.pdf	

Tabla 3: P.003- Comprobación de la correcta creación del elemento teatro

Prueba	P.004		Caso de Uso		C.U.007	
Objetivo	Comprobar que un elemento del tipo Temporada se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Nombre		Accesible		
		Descripción		La descripción		
		Banner		t.jpg		
		Texto		El texto		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre		Accesible		{x =100}
		Banner		t.jpg		{x =40}
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner		t.jpg		t.pdf

Tabla 4: P.004- Comprobación de la correcta creación del elemento temporada

Prueba	P.005		Caso de Uso		C.U.009
Objetivo	Comprobar que un elemento del tipo Producción se crea correctamente				
Criticidad	Alta	X	Media		Baja
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Nombre	Accesible		
		Autor	Francisco		
		Cartel	t.jpg		
		Resumen	El resumen		
		Sinopsis	La sinopsis		
		Texto del cartel	El texto		
		Id Idioma	Español		
		Id Producción	Cortes		
		Banner	t.jpg		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible	{x =100}	
		Autor	Francisco	{x =100}	
		Productor	Cortes	{x =100}	
		Director	Mena	{x =100}	
		Cartel	t.jpg	{x =40}	
		Video	El video	{x =150}	
		Banner	t.jpg	{x =40}	
Numérico	[Campo] debe ser numérico	Duración	2.25	Dos	
		N. de audiodescrip.	30	Treinta	
		N. de subtítulos	3	Tres	
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner	t.jpg	t.pdf	
		Cartel	t.jpg	t.pdf	

Tabla 5: P.005- Comprobación de la correcta creación del elemento producción

Prueba	P.006		Caso de Uso			C.U.0011
Objetivo	Comprobar que un elemento del tipo Performance se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Id Temporada	Accesible			
		Id Teatro	Teatro C.M.C			
		Id Producción	Producto			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Subtitulado	Info. Subtitulos	{x =2000}		
		Lengua de signos	Signos	{x =2000}		
		Audiodescripcion	Audios	{x =2000}		
		Sonido	Sonido de sala	{x =2000}		
		Bucle magnético	Bucle	{x =2000}		
		Descuentos	Información	{x =2000}		
		Enlace	www.link.es	{x =150}		
		Grupos	Información	{x =2000}		
Numérico	[Campo] debe ser numérico	Id Temporada	Accesible	Dos		
		Id Teatro	Teatro C.M.C	Treinta		
		Id Producción	Producto	Tres		

Tabla 6: P.006- Comprobación de la correcta creación del elemento performance

Prueba	P.007		Caso de Uso		C.U.014
Objetivo	Comprobar que un elemento del tipo Enlace se crea correctamente				
Criticidad	Alta	X	Media		Baja
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/off		
		Enlace	www.link.es		
		Imagen	t.jpg		
		Resumen	El resumen		
		Texto	El texto		
		Id enlace	El texto		
		Id Idioma	Español		
		Nombre	Cortes		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible	{x =200}	
		Imagen	t.jpg	{x =40}	
Numérico	[Campo] debe ser numérico	Orden	1	Uno	
		Id Idioma	30	Treinta	
		Id enlace	3	Tres	
Tipo de archivo	[Campo] debe ser jpg, gif o png	Imagen	t.jpg	t.pdf	

Tabla 7: P.007- Comprobación de la correcta creación del elemento enlace

Prueba	P.008		Caso de Uso		C.U.0020	
Objetivo	Comprobar que un elemento del tipo Noticias se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Visible		On/Off		
		Id Idioma		Español		
		Titulo		El titulo		
		Resumen		El resumen		
		Texto		El texto		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Titulo		El titulo		{x =200}

Tabla 8: P.008- Comprobación de la correcta creación del elemento noticias



Prueba	P.009		Caso de Uso			C.U.0023
Objetivo	Comprobar que un elemento del tipo Portada se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo	Ejecución			
Realizada	Mensaje		Correcta		Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Imagen	Accesible			
		Enlace	www.link.es			
		Titulo	El titulo			
		Texto	El texto			
		Texto Alt.	El texto 2			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Imagen	t.jpg		{x =40}	
		Enlace	www.link.es		{x =100}	
		Titulo	El titulo		{x =200}	
		Texto	El texto		{x =2000}	
		Texto Alt.	El texto 2		{x =250}	

Tabla 9: P.009- Comprobación de la correcta creación del elemento portada

Prueba	P.010		Caso de Uso			C.U.0023
Objetivo	Comprobar que un elemento del tipo Galería se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo	Ejecución			
Realizada	Mensaje		Correcta		Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Nombre	Accesible			
		Texto	El texto			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible		{x =100}	

Tabla 10: P.0010- Comprobación de la correcta creación del elemento galería

Prueba	P.011		Caso de Uso		C.U.0027	
Objetivo	Comprobar que un elemento del tipo Idioma se crea correctamente					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Crear” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento creado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Nombre	Accesible			
		Imagen	t.jpg			
		Texto Alt.	El texto			
		Abreviatura	Es			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible		{x =40}	
		Texto Alt.	El texto		{x =40}	

Tabla 11: P.0011- Comprobación de la correcta creación del elemento idioma

Prueba	P.012		Caso de Uso			C.U.029
Objetivo	Verificar que al pulsar “logout” la sesión se cierra.					
Criticidad	Alta	X	Media		Baja	
Punto de medición	Desde que se hace clic en el botón “Logout” hasta que se devuelve la página de autenticación de usuario.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Verificar que aparece el formulario de autenticación de usuario.			Formulario de autenticación de usuario indicando los campos erróneos o incompletos o si se produce algún error inesperado.			

Tabla 12: P.0012-Verificación del cierre de sesión de la aplicación

Prueba	P.013		Caso de Uso		C.U.004	
Objetivo	Comprobar que un elemento del tipo Ciudad se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Nombre		Madrid		
		Visible		On/Off (*)		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre		Madrid		{x=100}

Tabla 13: P.0013- Comprobación de la correcta edición del elemento ciudad

Prueba	P.014		Caso de Uso			C.U.006
Objetivo	Comprobar que un elemento del tipo Teatro se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo	Ejecución			
Realizada	Mensaje		Correcta	Incorrecta		
Obligatorio	[Campo] es obligatorio	Nombre	Teatro C.M.C			
		Ciudad	Madrid			
		Dirección	Avd/Carlos III			
		Latitud	40.411320			
		Longitud	-3.708740			
		Historia	La historia			
		Banner	t.jpg			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Teatro C.M.C	{x =100}		
		Código Postal	28024	{x =5}		
		Dirección	Avd/Carlos III	{x =100}		
		Link	www.link.es	{x =150}		
		Historia	La historia	{x =2000}		
		Banner	t.jpg	{x =40}		
Numérico	[Campo] debe ser numérico	Código Postal	28025	Tres		
		Latitud	40.411320	Cuatro		
		Longitud	-3.708740	Cinco		
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner	t.jpg	t.pdf		

Tabla 14: P.0014- Comprobación de la correcta edición del elemento teatro

Prueba	P.015		Caso de Uso		C.U.008	
Objetivo	Comprobar que un elemento del tipo Temporada se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Nombre		Accesible		
		Descripción		La descripción		
		Banner		t.jpg		
		Texto		El texto		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre		Accesible		{x =100}
		Banner		t.jpg		{x =40}
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner		t.jpg		t.pdf

Tabla 15: P.0015- Comprobación de la correcta edición del elemento temporada

Prueba	P.016	Caso de Uso			C.U.010
Objetivo	Comprobar que un elemento del tipo Producción se modifica correctamente				
Criticidad	Alta		Media	X	Baja
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Nombre	Accesible		
		Autor	Francisco		
		Cartel	t.jpg		
		Resumen	El resumen		
		Sinopsis	La sinopsis		
		Texto del cartel	El texto		
		Id Idioma	Español		
		Id Producción	Cortes		
		Banner	t.jpg		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible	{x =100}	
		Autor	Francisco	{x =100}	
		Productor	Cortes	{x =100}	
		Director	Mena	{x =100}	
		Cartel	t.jpg	{x =40}	
		Video	El video	{x =150}	
		Banner	t.jpg	{x =40}	
Numérico	[Campo] debe ser numérico	Duración	2.25	Dos	
		N. de audiodescrip.	30	Treinta	
		N. de subtítulos	3	Tres	
Tipo de archivo	[Campo] debe ser jpg, gif o png	Banner	t.jpg	t.pdf	
		Cartel	t.jpg	t.pdf	

Tabla 16: P.0016- Comprobación de la correcta edición del elemento producción

Prueba	P.017		Caso de Uso			C.U.0012
Objetivo	Comprobar que un elemento del tipo Performance se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Id Temporada	Accesible			
		Id Teatro	Teatro C.M.C			
		Id Producción	Producto			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Subtitulado	Info. Subtitulos	{x =2000}		
		Lengua de signos	Signos	{x =2000}		
		Audiodescripcion	Audios	{x =2000}		
		Sonido	Sonido de sala	{x =2000}		
		Bucle magnético	Bucle	{x =2000}		
		Descuentos	Información	{x =2000}		
		Enlace	www.link.es	{x =150}		
		Grupos	Información	{x =2000}		
Numérico	[Campo] debe ser numérico	Id Temporada	Accesible	Dos		
		Id Teatro	Teatro C.M.C	Treinta		
		Id Producción	Producto	Tres		

Tabla 17: P.0017- Comprobación de la correcta edición del elemento performance

Prueba	P.018	Caso de Uso			C.U.015
Objetivo	Comprobar que un elemento del tipo Enlace se modifica correctamente				
Criticidad	Alta		Media	X	Baja
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/off		
		Enlace	www.link.es		
		Imagen	t.jpg		
		Resumen	El resumen		
		Texto	El texto		
		Id enlace	El texto		
		Id Idioma	Español		
		Nombre	Cortes		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible	{x =200}	
		Imagen	t.jpg	{x =40}	
Numérico	[Campo] debe ser numérico	Orden	1	Uno	
		Id Idioma	30	Treinta	
		Id enlace	3	Tres	
Tipo de archivo	[Campo] debe ser jpg, gif o png	Imagen	t.jpg	t.pdf	

Tabla 18: P.0018- Comprobación de la correcta edición del elemento enlace



Prueba	P.019		Caso de Uso		C.U.0021	
Objetivo	Comprobar que un elemento del tipo Noticias se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Visible		On/Off		
		Id Idioma		Español		
		Titulo		El titulo		
		Resumen		El resumen		
		Texto		El texto		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Titulo		El titulo		{x =200}

Tabla 19: P.0019- Comprobación de la correcta edición del elemento noticias

Prueba	P.020	Caso de Uso			C.U.0024
Objetivo	Comprobar que un elemento del tipo Portada se modifica correctamente				
Criticidad	Alta			X	Media
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.		
Juego de Datos					
Validación		Campo	Ejecución		
Realizada	Mensaje		Correcta	Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off		
		Imagen	Accesible		
		Enlace	www.link.es		
		Titulo	El titulo		
		Texto	El texto		
		Texto Alt.	El texto 2		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Imagen	t.jpg	{x =40}	
		Enlace	www.link.es	{x =100}	
		Titulo	El titulo	{x =200}	
		Texto	El texto	{x =2000}	
		Texto Alt.	El texto 2	{x =250}	

Tabla 20: P.0020- Comprobación de la correcta edición del elemento portada

Prueba	P.021		Caso de Uso		C.U.0024	
Objetivo	Comprobar que un elemento del tipo Galería se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo		Ejecución		
Realizada	Mensaje			Correcta		Incorrecta
Obligatorio	[Campo] es obligatorio	Visible		On/Off		
		Nombre		Accesible		
		Texto		El texto		
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre		Accesible		{x =100}

Tabla 21: P.0021- Comprobación de la correcta edición del elemento galería

Prueba	P.022	Caso de Uso				C.U.0028
Objetivo	Comprobar que un elemento del tipo Idioma se modifica correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Editar” hasta que se devuelve la página de detalles con los atributos guardados.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento editado			Aparece de nuevo el formulario de entrada, indicando los campos erróneos y la información del error, o una página de error si se produce algún problema inesperado.			
Juego de Datos						
Validación		Campo	Ejecución			
Realizada	Mensaje		Correcta		Incorrecta	
Obligatorio	[Campo] es obligatorio	Visible	On/Off			
		Nombre	Accesible			
		Imagen	t.jpg			
		Texto Alt.	El texto			
		Abreviatura	Es			
Longitud Max.	[Campo] no puede ser mayor de {x} caracteres	Nombre	Accesible		{x =40}	
		Texto Alt.	El texto		{x =40}	

Tabla 22: P.0022- Comprobación de la correcta edición del elemento idioma

Prueba	P.023		Caso de Uso			C.U.016
Objetivo	Comprobar que un elemento del tipo Enlace se elimina correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Eliminar” hasta que se devuelve la página con el listado de elementos de la base de datos.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparecen dos confirmaciones para eliminar el elemento. Tras aceptarlas aparece la página con la lista de elementos que quedan en la base de datos			Aparece una página de error si se produce algún problema inesperado.			

Tabla 23: P.0023- Comprobación de la correcta eliminación del elemento enlace

Prueba	P.024		Caso de Uso			C.U.0022
Objetivo	Comprobar que un elemento del tipo Noticias se elimina correctamente					
Criticidad	Alta		Media	X	Baja	
Punto de medición	Desde que se hace clic en el botón “Eliminar” hasta que se devuelve la página con el listado de elementos de la base de datos.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparecen dos confirmaciones para eliminar el elemento. Tras aceptarlas aparece la página con la lista de elementos que quedan en la base de datos			Aparece una página de error si se produce algún problema inesperado.			

Tabla 24: P.0024- Comprobación de la correcta eliminación del elemento noticia

Prueba	P.025	Caso de Uso			C.U.0025
Objetivo	Comprobar que un elemento del tipo Portadas se elimina correctamente				
Criticidad	Alta		Media	X	Baja
Punto de medición	Desde que se hace clic en el botón “Eliminar” hasta que se devuelve la página con el listado de elementos de la base de datos.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparecen dos confirmaciones para eliminar el elemento. Tras aceptarlas aparece la página con la lista de elementos que quedan en la base de datos			Aparece una página de error si se produce algún problema inesperado.		

Tabla 25: P.0025- Comprobación de la correcta eliminación del elemento portada

Prueba	P.026	Caso de Uso			C.U.0019
Objetivo	Comprobar que un elemento del tipo Galería se elimina correctamente				
Criticidad	Alta		Media	X	Baja
Punto de medición	Desde que se hace clic en el botón “Eliminar” hasta que se devuelve la página con el listado de elementos de la base de datos.				
Contenido a chequear					
Ejecución correcta			Ejecución incorrecta		
Aparecen dos confirmaciones para eliminar el elemento. Tras aceptarlas aparece la página con la lista de elementos que quedan en la base de datos			Aparece una página de error si se produce algún problema inesperado.		

Tabla 26: P.0026- Comprobación de la correcta eliminación del elemento galería

Prueba	P.027	Caso de Uso			-	
Objetivo	Comprobar que al pulsar el id de un elemento muestra los detalles de éste					
Criticidad	Alta		Media		Baja	X
Punto de medición	Desde que se hace clic en el id hasta que se devuelve la página de detalles del elemento seleccionado.					
Contenido a chequear						
Ejecución correcta			Ejecución incorrecta			
Aparece la página de detalles del elemento seleccionado			Aparece una página de error si se produce algún problema inesperado.			

Tabla 27: P.0027- Demostración de los detalles de un elemento al pulsar el id